N73- 26884

SHUTTLE

LMSC-A991396
30 JUNE 1973

FINAL REPORT

SHUTTLE CRYOGENICS SUPPLY SYSTEM OPTIMIZATION STUDY

VOLUME V B-2

APPENDIX TO PROGRAMMERS MANUAL FOR MATH MODEL PART 1

CONTRACT NAS9-11330

Prepared for Manned Spacecraft Center
by
Manned Space Programs, Space Systems Division

LOCKHEED MISSILES & SPACE COMPANY. INC.
A SUBSIDIARY OF LOCKHEED AIRCRAFT CORPORATION

DRL Line Item 9
DRD No. MA-129T

LMSC-A991396

FINAL REPORT
SHUTTLE CRYOGENIC SUPPLY SYSTEM
OPTIMIZATION STUDY

VOLUME VB-2

APPENDIX TO PROGRAMMERS MANUAL FOR MATH MODEL

PART 1

Contract NAS 9-11330

Prepared for Manned Spacecraft Center By
Manned Space Programs, Space Systems Division

LOCKHEED MISSILES & SPACE COMPANY

FOREWORD

This Final Report provides the results obtained in the Shuttle Cryogenics Supply System Optimization Study, NAS 9-11330, performed by Lockheed Missiles & Space Company (LMSC) under contract to the National Aeronautics and Space Administration, Manned Spacecraft Center, Houston, Texas. The study was under the technical direction of Mr. T. L. Davies, Cryogenics Section of the Power Generation Branch, Propulsion and Power Division. Technical effort producing these results was performed in the period from October 1970 to June 1973.

The Final Report is published in eleven volumes*:

Volume I	Executive Summary
Volumes II, III, and IV	Technical Report
Volumes VA-1 and VA-2	Math Model – Users Manual
Volumes VB-1, VB-2, VB-3, and VB-4	Math Model – Programmers Manual
Volume VI	Appendixes

The LMSC Staff participants are as follows:

Study Manager	L. L. Morgan
Subsystem Evaluations	C. J. Rudey D. P. Burkholder C. F. Merlet W. H. Brewington
Integrated Systems	H. L. Jensen
Component Analyses	B. R. Bullard F. L. Bishop
Thermodynamics	G. E. Heuer R. M. Vernon J. Gries D. R. Elgin

*The Table of Contents for all volumes appears in Volume I only. Section 12 in Volume III contains the List of References for Volumes I through IV.

Thermal Protection	G. E. Heuer R. Cima
Fluid Dynamics	D. P. Burkholder R. Cima
Propellant Acquisition	M. P. Hollister R. K. Grove
Design	R. A. Michael
Structural Analysis	M. L. Vaughn C. C. Richie
Instrumentation	R. R. Gaura
Reusability/Reliability	R. F. Hausman
Failure Modes and Effect Analyses	D. C. Saunders
Requirements and Criteria	C. F. Merlet
Safety and Mission Completion	C. F. Merlet
Math Model	R. F. Hausman J. McKay

<u>Cryogenic Cooling Subtask</u>

Subsystem Evaluation	H. L. Jensen
Component Analyses	G. Heuer AiResearch
Thermodynamics	R. Cima
Thermal Protection	G. E. Heuer

CONTENTS

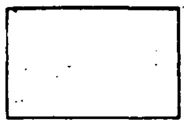
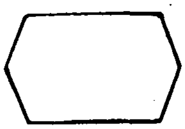
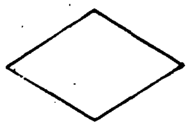
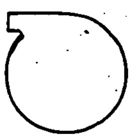
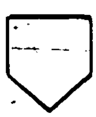
THE CRYOGENIC INTEGRATED
MATH MODEL PROGRAM
(TCIMM)

APPENDIX - A

FLOW CHART SYMBOLS

APPENDIX A

FLOW CHART SYMBOLS

SYMBOL	DEFINITION
	SUBPROGRAM REFERENCE
	PROCESSING FUNCTION
	PREPARATION FUNCTION
	DECISION FUNCTION
	TAPE OR FILE INPUT OR OUTPUT
	CARD INPUT
	FLOW DIRECTOR
	OFF-PAGE STEP CONNECTOR

Appendix B
THE CRYOGENIC INTEGRATED MATH MODEL (TCIMM)

PART I – PROGRAM LISTING

The program listing presented in the following pages was produced using the EXEC-8 LISTALL processor which lists a file in alphabetical order. Since the processor does not differentiate between subroutines, functions and Procedure Definition Processors (PDPs), each subprogram has been relabeled to clearly identify the type of symbolic listing presented.

The alphabetical listing permits rapid list scanning when searching for a particular subprogram.

While symbolic listings are quite useful in understanding the coding of a particular subprogram, it is recommended that the program user create and maintain a standard compiler listing output file, since the additional information provided by the compiler is very useful in troubleshooting and debugging changes to the basic coding.

The program list file follows:

******** SUBRØUTINE ACCRES

```
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C                    * ROUTINE NAME - ACCUMULATOR RESIDUALS           *
C                    *                DETERMINATION ROUTINE           *
C                    * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                    * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                    * DATE CODED   - 5/15/70                         *
C                    * REVISED      - JULY  1972                      *
C                    * PROGRAMMER   - J. MCKAY  D1943  201  45178 *
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ACCRES
C
      INCLUDE CACCUM
C
      DO 100 IGS=1,2
C                    FIND DENSITY OF GAS  (RHOG)
      CALL GSDNST (IGS,ATEMP(IGS),APRES(IGS),RHOG)
C                    CALCULATE WEIGHT OF GAS RESIDUALS
      WGRACC(IGS) = RHOG*AVOL(IGS)*NAOP(IGS)
  100 CONTINUE
C
      RETURN
      END
```

******** SUBRØUTINE ACQWT

```
C                    * * * * * * * * * * * * ** * * * * * * * * * * *
C                    * ROUTINE NAME - ACQUISITION SYSTEM WEIGHTS     *
C                    *                DETERMINATION ROUTINE          *
C                    * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                    * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                    * DATE CODED   - 5/20/70                        *
C                    * REVISED      - JULY  1972                     *
C                    * PROGRAMMER   - J. MCKAY  D1943  201  45178 *
C                    * * * * * * * * * * * * ** * * * * * * * * * * *
C
      SUBROUTINE ACQWT
C
      INCLUDE CCNFIG
      INCLUDE CTANK
C
      DIMENSION CST(2,3)
C
      DATA (CST(I,1),I=1,2) / .0035 ,.0333   /
      DATA (CST(I,2),I=1,2) / .00235,.01     /
      DATA (CST(I,3),I=1,2) / .0025 ,.007429 /
C
C         ***** THERE ARE 3 TYPES OF ACQUISITION DEVICES.
C                   IDX = 1     SURFACE TENSION DEVICE
C                   IDX = 2     POSITIVE DISPLACEMENT DEVICE
C                   IDX = 3     DIELECTROPHORETIC DEVICE
C
      DO 1000 II = 1,2
      IDX = SATYPE(II)
C
      WTACQ(II,1) = CST(II,IDX) * WPTOT(II)
      IDX = INDXTK(II)
      WEIGHT(IDX) = WEIGHT(IDX) + WTACQ(II,1)
 1000 CONTINUE
C                   PRINT ACQUISITION SYSTEM WEIGHTS
      CALL OTPACQ
      RETURN
      END
```

******** SUBRØUTINE ALPHAB

```
      SUBROUTINE ALPHAB(DB,DIB,TMODB)
C
C     CALCULATES ISOTHERMAL BULK MODULUS FROM THE EQUATION OF STATE
C     IN BRITISH UNITS
C     MUST FOLLOW CALLS OF PROPB AND DPDDB TO DEFINE D AND DI
C     INPUT AND OUTPUT PARAMETERS ARE IN BRITISH UNITS
C
      TMODB=-DB*DIB
      RETURN
      END
```

******** SUBRØUTINE APUFLØ

```
      SUBROUTINE APUFLO
C
      INCLUDE CAPU
      INCLUDE CCNTRL
      INCLUDE CDCYCL
      INCLUDE CENG
      INCLUDE CIOUNT
      INCLUDE CMATRL
      INCLUDE  TABLOK
C
C  **********************************************************************
C
C  FIND PERCENT POWER FOR EACH DUTY CYCLE POINT
C
      TOTHPR = HPR*NAPU
      DO 200 IX = 1,KCYCLE
      PCTHP(IX) = (100.0 * NEOP(IX) * HP(IX))/TOTHPR
  200 CONTINUE
C
C  **********************************************************************
C
C  CALCULATE PROPELLANT TEMPERATURE AT APU GAS GENERATOR INLET
C
      IF(TIT - 2060.0) 205,205,210
  205 TPF = (1.17 - FMR)/.000563
      GO TO 215
  210 TPF = (1.27 - FMR)/.000556
C
C  **********************************************************************
C
C  SELECT THE PROPER COEFFICIENTS FOR COMPUTING THE REFERENCE PROPELLANT
C  FLOW RATE.
C
  215 IF(PGG - 900.0) 216,220,216
  216 IF(PGG - 600.0) 217,225,217
  217 IF(PGG - 300.0) 218,230,218
C
C  ERROR IN THE SPECIFICATION OF PGG.
C
  218 WRITE(6,25) PGG
   25 FORMAT(T10,'PGG IS SPECIFIED INCORRECTLY.  PGG = ',F8.2,//)
      CALL EXIT
  220 M = 0
      RRFP = 8.70
      GO TO 240
  225 M = 1
      RRFP = 9.30
      GO TO 240
  230 M = 2
      RRFP = 10.57
  240 CONTINUE
C
C  **********************************************************************
C
C  COMPUTE THE REFERENCE PROPELLANT FLOW RATE IN POUNDS PER MINUTE AND
C  THE CORRECTION FACTOR FOR THE REFERENCE PROPELLANT FLOW RATE WHEN
C  TIT = 2060 DEGREES R.
```

```
C
      C1 = 1.0 + FMR
      C2 = 1.0 + (1.0/FMR)
      WDT = 0.0
      WTBURN = 0.0
      TIPWT = 0.0
      WDOT  = 0.0
      I = 0
      DO 260 J1 = 1,NDCYCL,2
      I = I + 1
      IF(DCYCLE(J1))  ,260,
C
C     LOOK UP RR VS PCTHP
C
      CALL FINTAB (NTBID(3))
      XTAB(1) = PGG
      XTAB(2) = FMR
      XTAB(3) = PAMB(I)
      XTAB(4) = PCTHP(I)
      RR(I) = MIPE(4,XTAB)
C
C     CHECK TO SEE IF CORRECTION FACTOR IS NECESSARY
C
      IF(TIT - 2060.0) 250,245,250
C
C     LOOK UP CORRECTION FACTOR KK VS PCTHP
C
  245 CALL FINTAB (NTBID(4))
      KK(I) = MIPE(4,XTAB)
      RR(I) = RR(I) * KK(I)
  250 CONTINUE
C
C  ***********************************************************************
C
C  COMPUTE THE TOTAL PROPELLANT FLOW RATE OVER EACH CONSTANT POWER TIME
C  INTERVAL IN LBS/MIN, AND COMPUTE THE TOTAL PROPELLANT USED OVER THE
C  ENTIRE DUTY CYCLE.
C
      WD(I) = (TOTHPR * RR(I))/300.0
      WDOT1 = WD(I)
      TIPWT = TIPWT + WDOT1 * DCYCLE(J1)
      WDOTJ(I,2) = (WDOT1/C1)/60.0
      WDOTJ(I,1) = WDOT1/60.0 - WDOTJ(I,2)
      IF(NEOP(I).LT.0) GO TO 260
      WDOT = AMAX1(WDOT,WDOT1)
      WTBURN = WTBURN + WD(I) * DCYCLE(J1)
C
C         SET CONTINGENCY QUANTITY OF TOTAL PROPELLANT FOR RESERVE
C
      WDT = 1.3 * WTBURN
C
C  CALCULATE THE WEIGHT RATE OF H2 AND O2 FLOWING DURING THE TIME PERIOD
C  THETA(I) IN LBS/MIN.
C
      WDRH(I) = WD(I)/C1
      WDRO(I) = WD(I)/C2
  260 CONTINUE
C
```

```
********     APUFLO    ********

      C  CALCULATE THE TOTAL H2 AND O2 DELIVERED TO APU GAS GENERATOR IN LBS.
      C
            WDH = WDT/C1
            WDOTI(2) = (WDOT/C1)/60.0
      C
            WDO = WDT/C2
            WDOTI(1) = WDOT/60.0 - WDOTI(2)
      C
      C  ***********************************************************************
      C
      C  DETERMINE APU EXHAUST GAS TEMPERATURE DURING POWER TIME INTERVAL
      C  THETA(I).  FIRST COMPUTE THE SPECIFIC HEAT CAPACITY OF THE COMBUSTION
      C  PRODUCTS AS A FUNCTION OF MIXTURE RATIO AND TURBINE INLET TEMPERATURE
      C
            CALL CSUBP1(TIT,FMR,CP)
            DO 405 I = 1,KCYCLE
            TE(I) = TIT - ((42.42 * NEOP(I) * HP(I)) /( WD(I) * CP))
            TME = (TE(I) + TD)/2.0
      C
      C  COMPUTE THE SPECIFIC HEAT OF COMBUSTION PRODUCTS AS A FUNCTION OF
      C  MIXTURE RATIO AND TEMPERATURE OF EXHAUST GASES.
      C
            CALL CSUBP1(TME,FMR,CPE(I))
            D(I) = CPE(I)*(TE(I) - TD)
        405 CONTINUE
      C
      C  ***********************************************************************
      C
      C     *****  OUTPUT THE DATA  *****
      C
            CALL OPAPUF (KCYCLE)
      C
            RETURN
      C
            END
```

******** SUBRØUTINE APUSUB

```
      SUBROUTINE APUSUB
C
C
      INCLUDE CACCUM
      INCLUDE CAPU
      INCLUDE CDCYCL
      INCLUDE CENG
      INCLUDE CIOUNT
      INCLUDE CPUMP
      INCLUDE CTANK
C
C   ************************************************************************
C
C   BEGIN COMPUTATIONS FOR SUBCRITICAL STORAGE
C
C   ************************************************************************
C
C   SET SELECTED VARIABLES FROM INPUT DATA
C
      TAH = ATEMP(2)
      TAO = ATEMP(1)
      TSTH = SITEMP(2,1)
      TSTO = SITEMP(1,1)
      PSTH = SOPRES(2,1)
      PSTO = SOPRES(1,1)
C
C   ************************************************************************
C
C   SIZE HEAT EXCHANGER BETWEEN H2 ACCUMULATOR AND APU GAS GENERATOR
C
      CALL CSUBP(TPF,PGG,2,CPSAH)
      CALL CSUBP(TPF,PGG,1,CPSAO)
      C1 = CPSAH*(TPF - TAH)
      C2 = CPSAO*(TPF - TAO)
      DO 610 I = 1,KCYCLE
      Q4HDOT(I) = WDRH(I)*C1
      Q6ODOT(I) = WDRO(I)*C2
      WDG(I) = Q4HDOT(I)/D(I)
  610 WDJ(I) = Q6ODOT(I)/D(I)
C
C   ************************************************************************
C
C
C   COMPUTE THE TEMPERATURES OF THE EXHAUST GASES FROM THE H2 AND O2
C   CONDITIONING GAS GENERATORS.
C
      TGGCH = (MRGGCH + .00056*TAH + .056)*1000.0/.591
      TGGCO = (MRGGCO + .00056*TAH + .056)*1000.0/.591
C
C   ************************************************************************
C
C   SET THE TEMPERATURE OF THE GASES AT DISCHARGE FROM THE PUMP EQUAL TO
C   THE TEMPERATURE OF THE GAS IN THE STORAGE TANK.
C
      TPDH = TSTH
      TPDO = TSTO
C
```

```
C  ***********************************************************************
C
C  SIZE H2 AND O2 HEAT EXCHANGER BETWEEN THE PUMPS AND ACCUMULATORS
C
      HAH = HYENTH(PPDCH(2),TAH)
      HPH = HYENTH(PPDCH(2),TPDH)
      HAO = OXENTH(PPDCH(1),TAO)
      HPO = OXENTH(PPDCH(1),TPDO)
      TME = (TGGCO + TDGGO)/2.0
      CALL CSUBP1(TME,MRGGCO,CPGGO)
      C1 = (1.0 + MRGGCO)*CPGGO*(TGGCO - TDGGO)
      A1 = ((HAO - HPO)*MRGGCO)/C1
      A2 = (HAO - HPO)/C1
      TME = (TGGCH + TDGGH)/2.0
      CALL CSUBP1(TME,MRGGCH,CPGGH)
      C2 = (1.0 + MRGGCH)*CPGGH*(TGGCH - TDGGH)
      A4 = (HAH - HPH)/C2
      A3 = A4*MRGGCH
      C1 = (A1 + A4) + (A2*A3) - (A1*A4) - 1.0
      C2 = (A1*A4) - (A2*A3) - A4
      C3 = (A1*A4) - (A2*A3) - A1
      DO 650 I = 1,KCYCLE
      WGGH(I) = (WDRH(I)*C2 - WDRO(I)*A2)/C1
      WGGO(I) = (WDRO(I)*C3 - WDRH(I)*A3)/C1
      Q5HDOT(I) = (WDRH(I) + WGGH(I))*(HAH - HPH)
      Q7ODOT(I) = (WDRO(I) + WGGO(I))*(HAO - HPO)
  650 CONTINUE
C
C  ***********************************************************************
C
C  COMPUTE THE TOTAL WEIGHT OF H2 AND O2 NEEDED FOR THE H2 AND O2
C  CONDITIONING GAS GENERATORS.
C
      WTGGH = 0.0
      WTGGO = 0.0
      I = 0
      DO 660 II = 1,NDCYCL,2
      I = I + 1
      WTGGH = WTGGH + WGGH(I) * DCYCLE(II)
      WTGGO = WTGGO + WGGO(I) * DCYCLE(II)
  660 CONTINUE
C
C  ***********************************************************************
C
C  CALCULATE THE VOLUMES OF THE H2 AND O2 STORAGE TANKS.
C
      CALL RHOLIQ(TSTH,2,RHOLH)
      CALL RHOLIQ(TSTO,1,RHOLO)
C
C
      CALL GSDNST (2,TSTH,PSTH,RHOGH)
      CALL GSDNST (1,TSTO,PSTO,RHOGO)
C
C     * * *
C     WEIGHT OF H2 REQD. TO ABSORB H2 TANK HEATLEAK
C     * * *
C
      TAUSUM =0.0
```

```
      DO 661 II = 2,NDCYCL,2
      IF(DCYCLE(II)) 661,,
      TAUSUM = TAUSUM + DCYCLE(II+1)
  661 CONTINUE
      TAUSUM = TAUSUM/60.
      VESTH = ((WDH + WTGGH)/(RHOLH-RHOGH))*1.03
      AREATH = 4.84 * (VESTH**.667)
      TSATLV = PSATH(PSTH,HG,HL)
      DELHV  = (HG - HL)
      CALL TCOND(TENV,TSTH,SNBAR(2),SITHIK(2,1),SITYPE(2,1),QTHDOT)
      WSVHH = (1.0/DELHV)*(QTHDOT*AREATH*TAUSUM)
C     * * *
C     WEIGHT OF H2 REQD. TO ABSORB O2 TANK HEATLEAK
C     * * *
      DELTH2 = (TSTO - TSTH)
      CALL CSUBP(TSTO,PSTO,2,CPH1)
      CALL CSUBP(TSTH,PSTH,2,CPH2)
      AVGCPH = (CPH1 + CPH2)/2.0
      DELHDT = DELTH2 * AVGCPH
      VESTO = ((WDO + WTGGO)/(RHOLO-RHOGO))*1.03
      AREATO = 4.84 * (VESTO**.667)
      CALL TCOND(TENV,TSTO,SNBAR(1),SITHIK(1,1),SITYPE(1,1),QTODOT)
      WSVHO = (1.0/DELHDT)*(QTODOT*AREATO*TAUSUM)
      IF(WSVHH.GE.WSVHO) WSVH = WSVHH
      IF(WSVHH.LT.WSVHO) WSVH = WSVHO
C
      VSTH = ((WDH+WTGGH+WSVH)/(RHOLH-RHOGH))*1.03
      AREATH = 4.84 * (VSTH**0.667)
      VSTO = ((WDO+WTGGO)/(RHOLO-RHOGO))*1.03
C
      WPTOT(2) = WDH + WTGGH + WSVH
      WPTTH = WPTOT(2)
      WPTOT(1) = WDO + WTGGO
      WPTTO = WPTOT(1)
C
C    ***********************************************************************
C
C    CALCULATE THE WIEGHT OF THE H2 AND O2 STORAGE TANK RESIDUAL
C    PROPELLANTS.
C
C
      WSRH = RHOGH * VSTH
      WSRO = RHOGO * VSTO
C
C    ***********************************************************************
C
C    CALCULATE THE WEIGHT OF H2 AND O2 ACCUMULATOR RESIDUAL PROPELLANTS.
C
      CALL ZFIND(TPF,PPDCH(2),2,ZSAH)
      CALL ZFIND(TAH,PGG,2,ZSAHE)
      ZSAO  = ZGET(TPF,PPDCH(1),1)
      ZSAOE = ZGET(TAO,PGG,1)
      C1 = (PPDCH(2)/ZSAH) - (PGG/ZSAHE)
      C2 = (PPDCH(1)/ZSAO) - (PGG/ZSAOE)
C                    CALC. CAPICITY OF ACCUM.
      WSHB = (HPR*RRFP)/(18000.0*(1.0+FMR))
      WSOB = (HPR*RRFP)/(18000.0*(1.0+(1.0/FMR)))
C
```

```
*********    APUSUB    *********

            WRSAH = (WSHB*PGG)/(C1*ZSAHE)
            WRSAO = (WSOB*PGG)/(C2*ZSAOE)
      C
      C  ************************************************************************
      C
      C        *****  ACCUMULATOR VOLUME IS INPUT  *****
      C
      C  ************************************************************************
      C
      C  ************************************************************************
      C
      C        OUTPUT THE DATA
      C
      C  ************************************************************************
      C
            CALL OAPUSB (KCYCLE)
      C
            WRITE (IOT,999)
        999 FORMAT(/////T30,'*****  THE APU SUBCRITICAL CALCULATIONS HAVE BEEN
           1 COMPLETED  *****')
      C
      C
        667 CONTINUE
      C
            RETURN
      C
            END
```

********* SUBRØUTINE APUSUP

```
              SUBROUTINE APUSUP
      C
      C
            REAL  K7, K8, K11
      C
              INCLUDE CACCUM
              INCLUDE CAPU
              INCLUDE CCNTRL
              INCLUDE CDCYCL
              INCLUDE CENG
              INCLUDE CFUEL
              INCLUDE CHEX
              INCLUDE CIOUNT
              INCLUDE CMATRL
              INCLUDE CONST
              INCLUDE CPUMP
              INCLUDE CTANK
              INCLUDE TABLOK
      C
            DIMENSION WDOTX(MHX,2)
      C
            EQUIVALENCE (WVHO2,WVIHO), (WVHH2,WVIHH)
            EQUIVALENCE (WDOTCF,WDOTX)
      C
      C  ********************************************************************
      C
      C     BEGIN COMPUTATIONS FOR SUPERCRITICAL STORAGE
      C
      C  ********************************************************************
      C
      C  SET SELECTED VARIABLES FROM INPUT DATA
      C
              TAH = ATEMP(2)
              TAO = ATEMP(1)
              PCH = SOPRES(2,1)
              PCO = SOPRES(1,1)
      C
      C  ********************************************************************
      C
      C  SIZE H2 HEAT EXCHANGER BETWEEN H2 ACCUMULATOR AND APU GAS GENERATOR.
      C
        406 HPH = HYENTH(PCH,TPF)
            HAH = HYENTH(PCH,TAH)
            HPO = OXENTH(PCO,TPF)
            HAO = OXENTH(PCO,TAO)
            DO 410 I = 1,KCYCLE
            Q1HDOT(I) = WDRH(I)*(HPH - HAH)
            WDA(I) = Q1HDOT(I)/D(I)
      C
      C  ********************************************************************
      C
      C
      C  SIZE O2 HEAT EXCHANGER BETWEEN O2 ACCUMULATOR AND APU GAS GENERATOR.
      C
            Q1ODOT(I) = WDRO(I)*(HPO - HAO)
            WDD(I) = Q1ODOT(I)/D(I)
        410 CONTINUE
```

```
*********    APUSUP    *********
      C
      C
      C  *******************************************************************
      C
      C
      C         DETERMINE INITIAL TANK TEMPERATURES
      C
            CALL FINTAB(NTBID(8))
            XTAB(1) = PCO
            XTAB(2) = 70.126
            TEMPO2 = MIPE(2,XTAB)
      C
            CALL FINTAB(NTBID(7))
            XTAB(1) = PCH
            XTAB(2) = 4.365
            TEMPH2 = MIPE(2,XTAB)
      C
      C         DETERMINE INITIAL CSUBV VALUES FOR TANK T AND P CONDITIONS
      C
            CISBVO = CSUBV(TEMPO2,PCO,1)
            CISBVH = CSUBV(TEMPH2,PCH,2)
      C
      C  COMPUTE THE COMPRESSIBILITY OF H2 AT TEMPERATURE TFH AND PRESSURE PFH
      C
      C  COMPUTE THE COMPRESSIBILITY OF O2 AT TEMPERATURE TFO AND PRESSURE PFO
      C
            TTTH = 0.0
            TTTO = 0.0
            TTH2WD(1) = 0.0
            TTO2WD(I) = 0.0
            I = 0
            DO 450 II = 1,NDCYCL,2
            I = I + 1
      C
            CALL ZFIND(TFH,PFH,2,ZFH)
      C
            ZFO = ZGET(TFO,PFO,1)
      C
      C  COMPUTE THE PERCENT OF USABLE H2 AND O2 WITHDRAWN UP TO THIS POINT
      C  IN THE MISSION
      C
            TTTH = TTTH + DCYCLE(II) * WDRH(I)
            TTH2WD(I) = TTTH
            TTTO = TTTO + DCYCLE(II) * WDRO(I)
            TTO2WD(I) = TTTO
            PCH2WD(I) = TTH2WD(I)/WDH
            PCO2WD(I) = TTO2WD(I)/WDO
      C
      C  COMPUTE THE DENSITY OF H2 AND O2 AS A FUNCTION OF PERCENT WITHDRAWN.
      C
            C1 = 1.0 - ((0.0427*PFH)/(ZFH*TFH))
            C2 = 1.0 - ((0.04253*PFO)/(ZFO*TFO))
            RHOCH2(I) = 4.365 * (1.0 - (PCH2WD(I) * C1))
            RHOCO2(I) = 70.126 * (1.0 - (PCO2WD(I) * C2))
      C
      C  *******************************************************************
      C
      C  COMPUTE THE TEMPERATURE OF H2 IN STORAGE TANK DURING TIME INTERVAL
```

```
C     THETA(I) AS A FUNCTION OF DENSITY AND CONDITIONED PRESSURE.
C
          CALL FINTAB (NTBID(7))
          XTAB(1) = PCH
          XTAB(2) = RHOCH2(I)
          TTH(I) = MIPE(2,XTAB)
C
C     COMPUTE THE TEMPERATURE OF O2 IN STORAGE TANK AS A FUNCTION OF
C     DENSITY AND PRESSURE.
C
          CALL FINTAB (NTBID(8))
          XTAB(1) = PCO
          XTAB(2) = RHOCO2(I)
          TTO(I) = MIPE(2,XTAB)
C
C         ******************************************************************
C
C     COMPUTE SPECIFIC HEAT OF H2 AS A FUNCTION OF DENSITY AND STORED
C     PRESSURE.
C
          CALL FINTAB (NTBID(5))
          XTAB(1) = PCH
          XTAB(2) = RHOCH2(I)
          DQODWH(I) = MIPE(2,XTAB)
C
C     COMPUTE THE SPECIFIC HEAT OF O2 AS A FUNCTION OF DENSITY AND PRESSURE
C
          CALL PHTHON(TTO(I),RHOCO2(I),1,PHI,THETA)
          DQODWO(I) = THETA
C
C     **********************************************************************
C
C
C     COMPUTE THE REQUIRED FLOW RATE OF EXHAUST GASES THROUGH HEAT EXCHANGR
C     BETWEEN H2 TANK AND H2 ACCUMULATOR.
C
          HTH(I) = HYENTH(PCH,TTH(I))
          Q2HDOT(I) = WDRH(I) * ( HAH - HTH(I))
          WDB(I) = Q2HDOT(I)/D(I)
          HTO(I) = OXENTH(PCO,TTO(I))
C
C     DO THE SAME FOR THE HEAT EXCHANGER BETWEEN THE O2 TANK AND O2 ACCUMU-
C     LATOR.
C
          Q2ODOT(I) = WDRO(I) * ( HAO - HTO(I))
          WDE(I) = Q2ODOT(I)/D(I)
C
C     SIZE H2 TANK HEAT EXCHANGER AND O2 TANK HEAT EXCHANGER.
C
          Q3HDOT(I) = WDRH(I)*DQODWH(I)
          WDC(I) = Q3HDOT(I)/D(I)
          Q3ODOT(I) = WDRO(I)*DQODWO(I)
          WDF(I) = Q3ODOT(I)/D(I)
C
          CSUBVO(I) = CSUBV(TTO(I),PCO,1)
          CSUBVH(I) = CSUBV(TTH(I),PCH,2)
C
          CALL PHTHON(TTO(I),RHOCO2(I),1,PHI,THETA)
```

```
      PHIO2(I) = PHI
C
      CALL FINTAB (NTBID(40))
      XTAB(1) = PCH
      XTAB(2) = RHOCH2(I)
      PHIH2(I) = MIPE(2,XTAB)
C
  450 CONTINUE
C
C   ************************************************************************
C
C
C   CHECK ADEQUACY OF EXHAUST PRODUCTS FOR CONDITIONING THE GASES
C
      DO 500 I = 1,KCYCLE
      WSUM(I) = WDA(I) + WDB(I) + WDC(I) + WDD(I) + WDE(I) + WDF(I)
      DWDB(I) = 0.0
      WGHC(I) = 0.0
      WGOC(I) = 0.0
      DWDB(I) = WSUM(I) - WD(I)
  500 CONTINUE
C
      DO 501 I = 1,KCYCLE
      IF(WSUM(I) .LE. WD(I)) GO TO 501
      GO TO 460
  501 CONTINUE
      GO TO 470
C
C
C   ************************************************************************
C
C
C   CORRECT FOR THE EFFECTS OF USE OF THE SUPPLEMENTARY CONDITIONING GAS
C   GENERATOR AND HEAT EXCHANGER.
C
  460 DO 502 I = 1,KCYCLE
      THE = (TG + TD)/2.0
      CALL CSUBP1(THE,FMRG,CPG)
      K7 = CPG*(TG - TD)
      K8 = 1.0 + FMRG
      K11 = (WDC(I)/WDRH(I))+((WDE(I)*FMRG)/WDRO(I))+((WDF(I)*FMRG)
     1 /WDRO(I))
C
C   COMPUTE REFERENCE H2 AND O2 FLOWRATES TO SUPPLEMENTAL GAS GENERATOR.
C
      WGH(I) = (WD(I) - WSUM(I))/(K11 - ((K8*K7)/D(I)))
C
C
      WGHC(I) = WGH(I)
C
      WGOC(I) = FMRG*WGHC(I)
C
C   COMPUTE THE CORRECTED VALUES OF THE HEAT EXCHANGER EXHAUST REQUIREMEN
C   TS.
C
      WDBC(I) = WDB(I) - (WGHC(I)*((K8*K7)/D(I)))
      WDCC(I) = WDC(I) * (1.0 + (WGHC(I)/WDRH(I)))
      WDEC(I) = WDE(I) * (1.0 + (WGOC(I)/WDRO(I)))
```

```
          WDFC(I) = WDF(I) * (1.0 + (WGOC(I)/WDRO(I)))
      502 CONTINUE
C
          GO TO 471
C         ***
C         IF NO SUPPLEMENTARY CONDITIONING GAS GENERATOR AND HEAT EXCHANGER
C         ARE REQUIRED.
C         ***
      470 DO 503 I = 1,KCYCLE
          WDBC(I) = WDB(I)
          WDCC(I) = WDC(I)
          WDEC(I) = WDE(I)
          WDFC(I) = WDF(I)
      503 CONTINUE
C
      471 CONTINUE
C
C     COMPUTE THE CORRECTED TOTAL H2 AND O2 FLOWS TO THE H2 AND O2 ACCUMULA
C     TORS.
C
          DO 504 I = 1,KCYCLE
          WTH(I) = WDRH(I) + WGHC(I)
          WTO(I) = WDRO(I) + WGOC(I)
C
C     COMPUTE THE TOTAL ENTHALPY INCREMENT SUPPLIED BY SUPPLEMENTARY GAS.
C
          DELH(I) = WGHC(I) * K8 * K7
C
C     COMPUTE CORRECTED VALUES OF HEAT FLOW IN MAIN H2 AND O2 HEAT EXCHANG-
C     ERS BETWEEN TANK AND ACCUMULATOR.
C
          Q2HDTC(I) = D(I)*WDBC(I)
          Q2ODTC(I) = WTO(I)*(HAO - HTO(I))
C
C     COMPUTE THE CORRECTED VALUES OF THE HEAT FLOWS IN THE H2 AND O2 TANK
C     HEAT EXCHANGERS.
C
          Q3HDTC(I) = DQODWH(I)*WTH(I)
          Q3ODTC(I) = DQODWO(I)*WTO(I)
C
C            COMPUTE TEMPERATURE OF COLD FLUID ENTERING SUPPLEMENTARY GAS
C            GENERATOR
C
          CALL CSUBP(TAH,PCH,2,CPH(I))
C
          CALL CSUBP(TTH(I),PCH,2,CPBH(I))
C
          TSIN(I) = TAH - (((WGHC(I)+WGOC(I))*K7)/(WTH(I)*CPBH(I)))
C
      504 CONTINUE
C
          DO 505 I4 = 1,KCYCLE
          IF(TSIN(I4) .LT. TTH(I4)) GO TO 506
      505 CONTINUE
C
          GO TO 507
C
      506 CONTINUE
```

******** APUSUP ********

```
C
C         *** RESET VALUE OF FMR AND RECYCLE PROGRAM.
C
      FMR = FMR + 0.1
      LREPT = 1
C
      WRITE (IOT,7011) I4, TSIN(I4), FMR
 7011 FORMAT(///T20,100('*')/T20,100('*')//T27,'AN UNACCEPT
     1ABLE VALUE FOR TSIN HAS BEEN ENCOUNTERED - - - TSIN FOR CYCLE =',
     2I3,2X,'*',F8.2//T40,' THE APU FUEL MIXTURE RATIO HAS BEEN RESET T
     3O - - - FMR =',F5.2//T20,100('*')/T20,100('*')/T20,100('*'))
C
      RETURN
C
  507 CONTINUE
C
C ***********************************************************************
C
C
C
C  CALCULATE THE WEIGHT OF PROPELLANT TANK HEATER CIRCULATING COMPRESSOR
C  FIRST COMPUTE THE MAXIMUM FLOW RATE FOR HYDROGEN AND OXYGEN.
C
      DQWMXO = 0.0
      DQWMXH = 0.0
      DO 509 I = 1,KCYCLE
      DQWMXO = AMAX1(DQWMXO,DQODWO(I))
      DQWMXH = AMAX1(DQWMXH,DQODWH(I))
  509 CONTINUE
C
      WOMAX = 0.0
      WHMAX = 0.0
      DO 510 I = 1,KCYCLE
      WOMAX = AMAX1(WOMAX,WTO(I))
      WHMAX = AMAX1(WHMAX,WTH(I))
  510 CONTINUE
C
      DO 511 I = 1,KCYCLE
      QINTKO(I) = DQODWO(I) * WTO(I)
      QINTKH(I) = DQODWH(I) * WTH(I)
  511 CONTINUE
C
      QMXTKO = 0.0
      QMXTKH = 0.0
      DO 512 I = 1,KCYCLE
      QMXTKO = AMAX1(QMXTKO,QINTKO(I))
      QMXTKH = AMAX1(QMXTKH,QINTKH(I))
  512 CONTINUE
C
C
C  COMPUTE THE SPECIFIC HEAT AND DENSITY OF H2 AND O2 AT THE FINAL
C  TEMPERATURE AND PRESSURE
C
      CALL CSUBP(TFO,PFO,1,CPFO)
      CALL DENSON(TFO,PFO,1,RHOOF,ZEE)
C
      IGAS = 2
      CALL CSUBP(TFH,PFH,IGAS,CPFH)
      RHOHF = (144.0*PFH)/(ZFH*TFH*FINDR(IGAS))
```

```
C
      WDTCPO = QMXTKO/(CPFO * (TD-100.0-TFO))
      WDTCPH = QMXTKH/(CPFH * (TD-100.0-TFH))
C
      WOCOMP = (0.01455*DELPCP*WDTCPO)/RHOOF
      WCIRCP(1) = WOCOMP
      WHCOMP = (0.01455*DELPCP*WDTCPH)/RHOHF
      WCIRCP(2) = WHCOMP
C
C CALCULATE THE WEIGHT OF THE H2 AND O2 STORAGE TANKS. FIRST COMPUTE
C THE TOTAL WEIGHT OF H2 AND O2 NEEDED FOR THE SUPPLEMENTAL GAS
C GENERATOR.
C
      WTGH = 0.0
      WTGO = 0.0
      I = 0
      DO 520 II = 1,NDCYCL,2
      I = I + 1
      WTGH = WTGH + WGHC(I) * DCYCLE(II)
      WTGO = WTGO + WGOC(I) * DCYCLE(II)
  520 CONTINUE
C
C ********************************************************************
C
C
C BEGIN CALCULATING WEIGHT OF H2 VENTED DURING THE MISSION. FIRST
C COMPUTE THE APPROXIMATE VOLUME OF THE H2 AND O2 STORAGE TANKS.
C
      VTH = (WDH + WTGH)/(4.35 - RHOHF)
      VTO = (WDO + WTGO)/(70.0 - RHOOF)
      RH = CBRT((3.0*VTH)/12.566)
      RO = CBRT((3.0*VTO)/12.566)
C
C COMPUTE THE AREA OF THE H2 AND O2 TANKS ASSUMING THAT THEY HAVE
C SPHERICAL GEOMETRY.
C
      ATH = 12.566*(RH**2)
      ATO = 12.566*(RO**2)
C
C COMPUTE THE HEAT FLOW PER UNIT AREA PER UNIT TIME ACROSS THE SURFACE
C OF THE TANK MATERIALS.
C
      WVIHH = 0.0
      WVIHO = 0.0
      I = 0
      DO 550 II = 2,NDCYCL,2
      I = I + 1
C
C CALL SUBROUTINE TCOND TO RETURN QDOTT.
C
      CALL TCOND(TENV,TFH,SNBAR(2),SITHIK(2,I),SITYPE(2,I),QDOTH)
      DHH = (QDOTH * ATH * DCYCLE(II))/60.0
C
      CALL TCOND(TENV,TFO,SNBAR(1),SITHIK(1,I),SITYPE(1,I),QDOTO)
      DHO = (QDOTO * ATO * DCYCLE(II))/60.0
C
C COMPUTE THE SPECIFIC HEAT OF H2 AT T= TTH(I) AND P= PCH.
C
```

```
      CALL CSUBP(TTH(I),PCH,2,CPH)
      WVIHO = WVIHO + (DHO/(CPH*(TFO-TTH(I))))
C
C  COMPUTE THE WEIGHT OF THE H2 IN THE TANK AT THE START OF THE COAST
C  PERIOD TAU(I).
C
      WH = WDH + WTGH - TTH2WD(I)
      C1 = (5.4*DHH)/(PCH*VTH)
      C2 = SQRT(1.0 + C1**2)
      WVIHH = WVIHH + (WH * (1.0+C1-C2))
  550 CONTINUE
C
C  COMPUTE RESULTANT TOTAL WEIGHT OF VENTED H2.
C
      IF(WVIHH-WVIHO) ,555,555
      WVH = WVIHO
      IF(WVH.LT.0.0) WVH = 0.0
      GO TO 560
  555 WVH = WVIHH
  560 CONTINUE
C
C  ******************************************************************
C
C
C  CALCULATE WEIGHT OF H2 AND O2 STORAGE TANKS.  FIRST COMPUTE THE
C  CORRECTED VOLUME OF THE H2 STORAGE TANK
C
      VTH = (WDH + WTGH + WVH)/(4.35 - RHOHF)
      MTRL = SMTYPE(2,1)
      CALL FINTAB (NTBID(9)+MTRL)
      FTUX = MIPE (1,TFH)
      ROFTUH = RHOL(MTRL)/1728./FTUX
      MTRL = SMTYPE
      CALL FINTAB (NTBID(9)+MTRL)
      FTUX = MIPE (1,TFO)
      ROFTUO = RHOL(MTRL)/1728./FTUX
      WGTHT = 7000.0*VTH*PCH*ROFTUH
      WGTOT = 7000.0*VTO*PCO*ROFTUO
C
      SMDIAM(2,1) = ((6.0 * VTH)/PI)**0.333
C
      SMDIAM(1,1) = ((6.0 * VTO)/PI)**0.333
C
C  ******************************************************************
C
C
C  CALCULATE WEIGHT OF RESIDUAL PROPELLANTS IN THE H2 AND O2 TANKS
C
      WRH = VTH*RHOHF
      WRO = VTO*RHOOF
C
C  ******************************************************************
C
C  CALCULATE WEIGHT OF H2 AND O2 ACCUMULATOR TANKS
C
      MTRL = AMTYPE(2)
      CALL FINTAB (NTBID(9)+MTRL)
      FTUX = MIPE (1,TAH)
```

```
      ROFTUH = RHOL(MTRL)/1728./FTUX
      MTRL = AMTYPE
      CALL FINTAB (NTBID(9)+MTRL)
      FTUX = MIPE (1,TAO)
      ROFTUO = RHOL(MTRL)/1728./FTUX
      CALL ZFIND(TAH,PCH,2,ZAH)
      CALL ZFIND(TAH,PFH,2,ZAHF)
      ZAO  = ZGET(TAO,PCO,1)
      ZAOF = ZGET(TAO,PFO,1)
      C1 = (PCH/ZAH) - (PFH/ZAHF)
      C2 = (PCO/ZAO) - (PFO/ZAOF)
C
      WAH = ((2.085/(1.0+FMR))*(PCH*ROFTUH)*(HPR*RRFP*TAH))/C1
C
      WAO=((0.1308/(1.0+(1.0/FMR)))*(PCO*ROFTUO)*(HPR*RRFP*TAO))/C2
C
C
C  CALCULATE THE WEIGHT OF THE H2 AND O2 ACCUMULATOR RESIDUAL PROPELLANT
C
      WSH = (HPR*RRFP)/(18000.0*(1.0+FMR))
      WSO = (HPR*RRFP)/(18000.0*(1.0+(1.0/FMR)))
      WRAH = (WSH*PFH)/(C1*ZAHF)
      WRAO = (WSO*PFO)/(C2*ZAOF)
C
C
C  ***********************************************************************
C
C  COMPUTE THE TOTAL PROPELLANT REQUIREMENT
C
C
      WPTOT(2) = WDH + WTGH + WVH
      WHTOT = WDH + WTGH + WVH
      WPTOT(1) = WDO + WTGO
      WOTOT = WDO + WTGO
C
C                    OUTPUT APU SUPERCRITICAL DATA
      CALL OAPUSP (KCYCLE)
C
C
C
C  ***********************************************************************
C     COMPUTE WEIGHT AND CHARACTERISTICS OF ALL HEAT EXCHANGERS
C  ***********************************************************************
C
      JX = 0
C
C     COMPUTE WEIGHT AND CHARACTERISTICS OF THE O2 HEAT EXCHANGER
C     BETWEEN THE O2 ACCUMULATOR AND THE APU
C
      JX = JX + 1
      JHX = JX
      IGAS = 1
C
      WDOMAX = 0.0
      DO 2010 I = 1,KCYCLE
      IF(WDRO(I).LT.WDOMAX) GO TO 2010
      WDOMAX = WDRO(I)
      IMAX = I
```

```
 2010 CONTINUE
C
      WDOTX(JX,IGAS) = 0.0166 * WDOMAX
      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
      HEXHIT(JX,IGAS) = TE(IMAX)
      HEXCIT(JX,IGAS) = TAO
      HEXHOT(JX,IGAS) = TD
      HEXCOT(JX,IGAS) = TPF
      HEXCOP(JX,IGAS) = PGG
      HXMRAT(JX,IGAS) = FMR
C
      CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS)
     1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
     2 HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOTH (JX,IGAS),
     3 WHXTOT(JX,IGAS))
C
      CALL GASGEN(JX,IGAS)
C
C     COMPUTE WEIGHT AND CHARACTERISTICS OF THE H2 HEAT EXCHANGER
C     BETWEEN THE H2 ACCUMULATOR AND THE APU
C
      IGAS = 2
C
      WDHMAX =0.0
      DO 2020 I = 1,KCYCLE
      IF(WDRH(I).LT.WDHMAX) GO TO 2020
      WDHMAX = WDRH(I)
      IMAX = I
 2020 CONTINUE
C
      WDOTX(JX,IGAS) = 0.0166 * WDHMAX
      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
      HEXHIT(JX,IGAS) = TE(IMAX)
      HEXCIT(JX,IGAS) = TAH
      HEXHOT(JX,IGAS) = TD
      HEXCOT(JX,IGAS) = TPF
      HEXCOP(JX,IGAS) = PGG
      HXMRAT(JX,IGAS) = FMR
C
      CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS)
     1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
     2 HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOTH (JX,IGAS),
     3 WHXTOT(JX,IGAS))
C
      CALL GASGEN(JX,IGAS)
C
C     COMPUTE WEIGHT AND CHARACTERISTICS OF THE H2 SUPPLEMENTARY GAS
C     HEAT EXCHANGER
C
      JX = JX + 1
      JHX = JX
      IGAS = 2
C
      CALL CSUBP1(TG,FMRG,CPG1)
      KB = CPG1*(TG-TD)
      WGHCMX = 0.0
      DO 2050 I = 1,KCYCLE
      IF(WGHC(I).LT.WGHCMX) GO TO 2050
```

```
********     APUSUP     ********

            WGHCMX = WGHC(I)
            IMAX = I
       2050 CONTINUE
      C
            QDTSMX = (WGHCMX+WGOC(IMAX)) * K8
      C
            TSINMN = TSIN(1)
            DO 2058 I = 2,KCYCLE
            IF(TSIN(I).EQ.0.0) GO TO 2058
            TSINMN = AMIN1(TSINMN,TSIN(I))
       2058 CONTINUE
      C
            WTHMAX = 0.0
            DO 2055 I = 1,KCYCLE
            IF(WTH(I).LT.WTHMAX) GO TO 2055
            WTHMAX = WTH(I)
            IMAX = I
       2055 CONTINUE
      C
            WDOTX(JX,IGAS) = 0.0166 * WTHMAX
            UCODE(JX,IGAS) = HXCODE(JX,IGAS)
            HEXHIT(JX,IGAS) = TG
            HEXCIT(JX,IGAS) = TSINMN
            HEXHOT(JX,IGAS) = TD
            HEXCOT(JX,IGAS) = TAH
            HEXCOP(JX,IGAS) = APRES(2)
            HEXCIP(JX,IGAS) = PCH - 25.0
            HXMRAT(JX,IGAS) = FMRG
      C
            IF(HEXCIT(JX,IGAS).EQ.HEXCOT(JX,IGAS)) GO TO 2025
      C
            CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS)
           1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
           2 HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOTH (JX,IGAS),
           3 WHXTOT(JX,IGAS))
      C
            CALL GASGEN(JX,IGAS)
      C
            GO TO 2026
      C
       2025 CONTINUE
      C
            WRITE (IOT,885) HXCODE(JX,IGAS)
        885 FORMAT(/T4,' *** HEAT EXCHANGER - ',A6, ' IS NOT IN USE IN THIS PR
           1OBLEM ***')
      C
       2026 CONTINUE
      C
      C     COMPUTE WEIGHT AND CHARACTERISTICS OF THE O2 HEAT EXCHANGER
      C     BETWEEN THE O2 STORAGE TANK AND THE ACCUMULATOR
      C
            JX = JX + 1
            JHX = JX
            IGAS = 1
      C
            WTOMAX = 0.0
            DO 2030 I = 1,KCYCLE
            IF(WTO(I).LT.WTOMAX) GO TO 2030
```

```
*********    APUSUP    *********
            WTOMAX = WTO(I)
            IMAX = I
 2030 CONTINUE
C
            WDOTX(JX,IGAS) = 0.0166 * WTOMAX
            UCODE(JX,IGAS) = HXCODE(JX,IGAS)
            HEXHIT(JX,IGAS) = TE(IMAX)
            HEXCIT(JX,IGAS) = TTO(IMAX)
            HEXHOT(JX,IGAS) = TD
            HEXCOT(JX,IGAS) = TAO
            HEXCOP(JX,IGAS) = PCO - HXCDLP(JX,IGAS)
            HEXCIP(JX,IGAS) = PCO
            HXMRAT(JX,IGAS) = FMR
C
            CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS)
          1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
          2 HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOTH (JX,IGAS),
          3 WHXTOT(JX,IGAS))
C
            CALL GASGEN(JX,IGAS)
C
C           COMPUTE WEIGHT AND CHARACTERISTICS OF THE H2 HEAT EXCHANGER
C           BETWEEN THE H2 STORAGE TANK AND THE ACCUMULATOR
C
            IGAS = 2
C
            WTHMAX = 0.0
            DO 2040 I = 1,KCYCLE
            IF(WTH(I).LT.WTHMAX) GO TO 2040
            WTHMAX = WTH(I)
            IMAX = I
 2040 CONTINUE
C
            WDOTX(JX,IGAS) = 0.0166 * WTHMAX
            UCODE(JX,IGAS) = HXCODE(JX,IGAS)
            HEXHIT(JX,IGAS) = TE(IMAX)
            HEXCIT(JX,IGAS) = TTH(IMAX)
            HEXHOT(JX,IGAS) = TD
            IF(DELH(IMAX).EQ.0.D) TSINMN = TAH
            HEXCOT(JX,IGAS) = TSINMN
            HEXCOP(JX,IGAS) = PCH - HXCDLP(JX,IGAS)
            HEXCIP(JX,IGAS) = PCH
            HXMRAT(JX,IGAS) = FMR
C
            CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS)
          1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
          2 HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOTH (JX,IGAS),
          3 WHXTOT(JX,IGAS))
C
            CALL GASGEN(JX,IGAS)
C
C           COMPUTE WEIGHT AND CHARACTERISTICS OF THE O2 HEAT EXCHANGER
C           OF THE O2 TANK CONDITIONING HEAT EXCHANGER
C
            JX = JX + 1
            JHX = JX
            IGAS = 1
C
```

```
      WDOTX(JX,IGAS) = 0.0166 * WDTCPO
      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
      HEXHIT(JX,IGAS) = TE(IMAX)
      HEXCIT(JX,IGAS) = TTO(IMAX)
      HEXHOT(JX,IGAS) = TD
      HEXCOT(JX,IGAS) = TD - 100.0
      HEXCIP(JX,IGAS) = PCO + DELPCP
      HEXCOP(JX,IGAS) = PCO
      HXMRAT(JX,IGAS) = FMR
C
      CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS)
     1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
     2 HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOTH (JX,IGAS),
     3 WHXTOT(JX,IGAS))
C
      CALL GASGEN(JX,IGAS)
C
C     COMPUTE WEIGHT AND CHARACTERISTICS OF THE H2 HEAT EXCHANGER
C     OF THE H2 TANK CONDITIONING HEAT EXCHANGER
C
      IGAS = 2
C
      WDOTX(JX,IGAS) = 0.0166 * WDTCPH
      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
      HEXHIT(JX,IGAS) = TE(IMAX)
      HEXCIT(JX,IGAS) = TTH(IMAX)
      HEXHOT(JX,IGAS) = TD
      HEXCOT(JX,IGAS) = TD - 100.0
      HEXCIP(JX,IGAS) = PCH + DELPCP
      HEXCOP(JX,IGAS) = PCH
      HXMRAT(JX,IGAS) = FMR
C
      CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS)
     1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
     2 HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOTH (JX,IGAS),
     3 WHXTOT(JX,IGAS))
C
      CALL GASGEN(JX,IGAS)
C
      CALL OTPHEX
C
C  ********************************************************************
C
      WRITE (IOT,996)
  996 FORMAT(/////T30,'*****  THE APU SUPERCRITICAL CALCULATIONS HAVE BE
     1EN COMPLETED  *****')
C
C  ********************************************************************
C
C  END OF SUPERCRITICAL COMPUTATIONS.
C
      RETURN
C
      END
```

******** SUBRØUTINE BETAB

```
      SUBROUTINE BETAB(TB,DB,NGAS,VEXB)
C
C     CALCULATES VOLUME EXPANSIVITY FROM EQUATION OF STATE IN BRITISH
C     UNITS
C
      T = TB
      D = DB
      M = 1
C
      IF(NGAS.EQ.1) KF = 1
      IF(NGAS.EQ.18) KF = 2
C
C     KF = 1    CALL IN OXYGEN PROPERTIES (DATAO2)
C     KF = 2    CALL IN NITROGEN PROPERTIES (DATAN2)
C
      IF(KF.EQ.1) CALL DATAO2
      IF(KF.EQ.2) CALL DATAN2
C
      D1 = DPDDB(T,D)
      D2 = DPDTB(T,D)
C
      VEXB = (1.0/D) * (D2/D1)
C
      RETURN
      END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - CCACUM

```
CACCUM* PROC
C
      PARAMETER NA2=2,KAC2=NA2+1,KAC3=2*NA2+1,KAC4=3*NA2+1,KAC5=4*NA2+1,
     1 KAC6=5*NA2+1,KAC7=6*NA2+1,KAC8=7*NA2+1,KAC9=8*NA2+1,KAC0=9*NA2
C
      INTEGER AITYPE,AMTYPE
C
      COMMON /CACCUM/  NAOP  (NA2),AMTYPE(NA2),AITYPE(NA2),ADIAM (NA2),
     1     AA     (NA2),AVOL  (NA2),ACWT  (NA2),AITHIK(NA2),ACIWT (NA2),
     2     AHFLUX(NA2),ANBAR (NA2),ANDELP(NA2),APRES (NA2),ATEMP (NA2),
     3     WGRACC(NA2),INDXAC(NA2),ACYHT (NA2)
C
      DIMENSION EQAC(KAC0)
C
      EQUIVALENCE           (EQAC      ,NAOP  ),(EQAC(KAC2),AMTYPE),
     1 (EQAC(KAC3),AITYPE),(EQAC(KAC4),ADIAM ),(EQAC(KAC5),AA    ),
     2 (EQAC(KAC6),AVOL  ),(EQAC(KAC7),ACWT  ),(EQAC(KAC8),AITHIK),
     3 (EQAC(KAC9),ACIWT )
C
C
C         ***** CACCUM VARIABLE DEFINITION
C             *
C             * AMTYPE - ACCUMULATOR MATERIAL TYPE (SEE CMATRL).
C             *
C             * AITYPE - ACCUMULATOR INSULATION TYPE (SEE CINSUL).
C             *
C             * ADIM   - DIMENSIONS OF THE ACCUMULATOR
C             *
C             * AIPRES - ACCUMULATOR INITIAL PRESSURE.
C             *
C             * AITEMP - ACCUMULATOR INITIAL TEMPRATURE.
C             *
C             * AHFLUX - ACCUMULATOR HEAT FLUX.
C             *
C             * AOPRES - ACCUMULATOR OPERATING PRESSURE.
C             *
C             * AVPRES - ACCUMULATOR VENTING PRESSURE.
C             *
C             * ANDELP - ACCUMULATOR NOMINAL OPERATING DELTA PRESSURE.
C             *
C             * AITHIK - ACCUMULATOR INSULATION THICKNESS.
C             *
C             * ANBAR  - NUMBER OF LAYERS OF INSULATION ON ACCUMULATOR
C             *
C             * NOTE --- EACH OF THE ABOVE   VARIABLES ARE SPECIFIED
C             *          FOR OXYGEN IN THE FIRST WORD AND HYDROGEN
C         *****          IN THE SECOND WORD.
C
  END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR - CAPU

```
CAPU* PROC
C
      PARAMETER LAPU = 20
C
      REAL KK, MRGGCH, MRGGCO
C
      COMMON /CIAPU/ M      ,NAPU  ,DELPCP,FMR   ,FMRG  ,HPR   ,MRGGCH,
     1 MRGGCO,PFH    ,PFO   ,PGG   ,RRFP  ,TD    ,TDGGH ,TDGGO ,TENV  ,
     2 TFH   ,TFO    ,TG    ,TIT   ,TME   ,TVH   ,TVO   ,EQAP1(5),
     3 EQAP2(11,2),EQAP3(11,2),LAPU1(4,8),LAPU2(4,11),LAPU3(4,11)
C
      COMMON /CVAPU/KK(LAPU), RR(LAPU), WD(LAPU), D (LAPU), TE(LAPU),
     1             CPE(LAPU),TTH(LAPU),TTO(LAPU),WDA(LAPU),WDB(LAPU),
     2             WDC(LAPU),WDD(LAPU),WDE(LAPU),WDF(LAPU),WDG(LAPU),
     3             WDJ(LAPU),WGH(LAPU),WTH(LAPU),WTO(LAPU),HTH(LAPU),
     4             HTO(LAPU),
     5             WDRH  (LAPU),WDRO  (LAPU), PCTHP (LAPU), WGGH  (LAPU)
     6            ,WGGO  (LAPU),WDFC  (LAPU), WGHC  (LAPU), WGOC  (LAPU)
     7            ,WDBC  (LAPU),WDCC  (LAPU), WDEC  (LAPU), DELH  (LAPU)
     8            ,DWDB  (LAPU),DQODWH(LAPU), DQODWO(LAPU), Q1HDOT(LAPU)
     9            ,Q1ODOT(LAPU),Q2HDOT(LAPU), Q2ODOT(LAPU), Q3HDOT(LAPU)
     T            ,Q3ODOT(LAPU),TTH2WD(LAPU), TTO2WD(LAPU), PCH2WD(LAPU)
     1            ,PCO2WD(LAPU),Q2HDTC(LAPU), Q2ODTC(LAPU), Q3HDTC(LAPU)
     2            ,Q3ODTC(LAPU),Q4HDOT(LAPU), Q5HDOT(LAPU), Q6ODOT(LAPU)
     3            ,Q7ODOT(LAPU),RHOCH2(LAPU), RHOCO2(LAPU), DQDWTH(LAPU)
     4            ,DQDWTO(LAPU),TSIN  (LAPU), CPH   (LAPU)
     5            ,CSUBVO(LAPU),CSUBVH(LAPU), PHIO2 (LAPU), PHIH2 (LAPU)
     6            ,QINTKO(LAPU),QINTKH(LAPU), CPBH  (LAPU), WSUM  (LAPU)
C
      EQUIVALENCE        (EQAP1    ,TPF    ),(EQAP1(2),WDO    ),
     1 (EQAP1(3),WDH    ),(EQAP1(4),WDT    ),(EQAP1(5),WTBURN)
     2,(EQAP2       ,WOCOMP),(EQAP2( 1,2),WHCOMP),(EQAP2( 2,1),WTGO   ),
     3 (EQAP2( 2,2),WTGH  ),(EQAP2( 3,1),VTO   ),(EQAP2( 3,2),VTH    ),
     4 (EQAP2( 4,1),ATO   ),(EQAP2( 4,2),ATH   ),(EQAP2( 5,1),WGTOT  ),
     5 (EQAP2( 5,2),WGTHT ),(EQAP2( 6,1),WVHO2 ),(EQAP2( 6,2),WVHH2  ),
     6 (EQAP2( 7,1),WRO   ),(EQAP2( 7,2),WRH   ),(EQAP2( 8,1),WAO    ),
     7 (EQAP2( 8,2),WAH   ),(EQAP2( 9,1),WSO   ),(EQAP2( 9,2),WSH    ),
     8 (EQAP2(10,1),WRAO  ),(EQAP2(10,2),WRAH  ),(EQAP2(11,1),WOTOT  ),
     9 (EQAP2(11,2),WHTOT )
C
      EQUIVALENCE             (EQAP3       ,TGGCO ),(EQAP3( 1,2),TGGCH ),
     1 (EQAP3( 2,1),WTGGO ),(EQAP3( 2,2),WTGGH ),(EQAP3( 3,1),AREATO),
     2 (EQAP3( 3,2),AREATH),(EQAP3( 4,1),VSTO  ),(EQAP3( 4,2),VSTH  ),
     3 (EQAP3( 5,1),QTODOT),(EQAP3( 5,2),QTHDOT),(EQAP3( 6,1),WSVHO ),
     4 (EQAP3( 6,2),WSVHH ),                     (EQAP3( 7,2),WSVH  ),
     5 (EQAP3( 8,1),WPTTO ),(EQAP3( 8,2),WPTTH ),(EQAP3( 9,1),WSRO  ),
     6 (EQAP3( 9,2),WSRH  ),(EQAP3(10,1),WSOB  ),(EQAP3(10,2),WSHB  ),
     7 (EQAP3(11,1),WRSAO ),(EQAP3(11,2),WRSAH )
C
C
C           *****
C               * NAPU  - NUMBER OF APU UNITS
C               * HPR   - HORSEPOWER RATING EACH APU
C               * FMR   - APU TURBINE MIXTURE RATIO
C               * PGG   - APU GAS GEN. INLET GAS PRESSURE
C               * TIT   - TURBINE INLET TEMP
C               * TD    - EXHAUST TEMP FOR HEAT EXCHANGER
```

```
********     CAPU     ********

C          * FMRG   - FUEL MIXTURE-RATIO FOR SUPPLEMENTAL GAS GEN.
C          * PFH    - FINAL H2 TANK PRESSURE
C          * PFO    - FINAL O2 TANK PRESSURE
C          * TFH    - FINAL H2 TANK TEMPERATURE
C          * TFO    - FINAL O2 TANK TEMPERATURE
C          * TG     - TEMP. OF EXHAUST PRODUCTS - SUPPL. GAS GEN.
C          * DELCP  - DELTA-P OF TANK CIRCULATING PUMP
C          * KK     - COR.FACTOR-REF.PROP.FLOW RATE FOR 2060 TIT.
C          * RR     - REF.PROP.FLOW RATE (LBS/MIN)
C          * WD     - DELIVERED FLOW RATE OVER INTERVAL (I)(LBS/MIN)
C          * D      - HEAT OF COMBUSTION PRODUCTS (BTU/LB)
C          * TE     - APU TURBINE EXHAUST TEMP.
C          * CPE    - EXHAUST SPEC.HT. AT(TE AND FMR)
C          * TTH    - TEMP. OF H2 IN TANK DURING INTERVAL(I)
C          * TTO    - TEMP. OF O2 IN TANK DURING INTERVAL(I)
C          * WDA    - APU EXHAUST FLOW THRU H2-HEX (ACCUM TO GAS GEN)
C          * WDB    - APU EXHAUST FLOW THRU H2-HEX (TANK TO ACCUM)
C          * WDC    - APU EXHAUST FLOW THRU H2-TANK HEX
C          * WDD    - APU EXHAUST FLOW THRU O2-HEX (ACCUM TO GAS GEN)
C          * WDE    - APU EXHAUST FLOW THRU O2-HEX (TANK TO ACCUM)
C          * WDF    - APU EXHAUST FLOW THRU O2-TANK HEX
C          * WDG    - APU EXHAUST FLOW THRU H2-HEX(SUBC.ACUM-GAS GEN)
C          * WDJ    - APU EXHAUST FLOW THRU O2-HEX(SUBC.ACUM-GAS GEN)
C          * WGH    - REF.H2 FLOWRATE TO SUPPLEMENTAL GAS GEN.
C          * WTH    - H2 FLOW TO ACCUM DURING INTERVAL (I)
C          * WTO    - O2 FLOW TO ACCUM DURING INTERVAL (I)
C          * HTH    - ENTHALPY OF H2 IN TANK AT INTERVAL(I)
C          * HTO    - ENTHALPY OF O2 IN TANK AT INTERVAL(I)
C          * WDRH   - WGT.RATE OF H2 FLOWING-INTERVAL(I), (LBS/MIN).
C          * WDRO   - WGT.RATE OF O2 FLOWING-INTERVAL(I), (LBS/MIN)
C          * PCTHP  - PERCENT HORSEPOWER REQD.-INTERVAL(I)
C          * WGGH   - H2 FLOW RATE TO COND.GAS GENS.-INTERVAL(I)
C          * WGGO   - O2 FLOW RATE TO COND.GAS GENS.-INTERVAL(I)
C          * WDFC   - CORRECTED WDF(I) FOR FLOW RATE CHANGE
C          * WGHC   - REF.H2 FLOW TO SUPPLEMENTAL GAS GEN.
C          * WGOC   - REF.O2 FLOW TO SUPPLEMENTAL GAS GEN.
C          * WDBC   - CORRECTED VALUES OF HEX EXHAUST REQMTS.(WDB(I))
C          * WDCC   - CORRECTED VALUES OF HEX EXHAUST REQMTS.(WDC(I))
C          * WDEC   - CORRECTED VALUES OF HEX EXHAUST REQMTS.(WDE(I))
C          * DELH   - TOTAL ENTHALPY INCREMENT FROM SUPPL.GAS GEN.
C          * DWDB   - REF.REDUCTION IN WDB FOR  APU EXHAUST AVAIL.
C          * TSIN   - T-COLD-IN FOR SUPPLEMENTARY GAS GENERATOR
C          * CPH    - SPECIFIC HEAT OF COLD FLUID INTO SUP.GAS GEN.
C          * DQODWH - HT.XFER.INTO O2 TANK DURING INTERVAL(I)
C          * DQODWO - HT.XFER.INTO H2 TANK DURING INTERVAL(I)
C          * Q1HDOT - HT.XFER.INTO HEX BETWEEN H2 ACCUM -APU GAS GEN.
C          * Q1ODOT - HT.XFER.INTO HEX BETWEEN O2 ACCUM -APU GAS GEN.
C          * Q2HDOT - HT.XFER.INTO HEX BETWEEN H2 TANK - H2 ACCUM.
C          * Q2ODOT - HT.XFER.INTO HEX BETWEEN O2 TANK - O2 ACCUM.
C          * Q3HDOT - HT.XFER.INTO H2 TANK - INTERVAL(I)
C          * Q3ODOT - HT.XFER.INTO O2 TANK - INTERVAL(I)
C          * Q2HDTC - CORRECTED VALUE OF Q2HDOT DUE TO SUPPL.GAS GEN.
C          * Q2ODTC - CORRECTED VALUE OF Q2ODOT DUE TO SUPPL.GAS GEN.
C          * Q3HDTC - CORRECTED VALUE OF Q3HDOT DUE TO SUPPL.GAS GEN.
C          * Q3ODTC - CORRECTED VALUE OF Q3ODOT DUE TO SUPPL.GAS GEN.
C          * Q4HDOT - HT.XFER.INTO HEX BETWEEN H2 ACCUM -APU GAS GEN.
C          * Q5HDOT - HT.XFER.INTO HEX BETWEEN H2 PUMP  - H2 ACCUM.
C          * Q6ODOT - HT.XFER.INTO HEX BETWEEN O2 ACCUM -APU GAS GEN.
```

```
*********     CAPU     *********

      C              * Q7ODOT - HT.XFER.INTO HEX BETWEEN O2 PUMP  - O2 ACCUM.
      C              * TTH2WD - TOTAL USABLE H2 WITHDRAWN TO END INTERVAL(I)
      C              * TTO2WD - TOTAL USABLE O2 WITHDRAWN TO END INTERVAL(I)
      C              * PCH2WD - PERCENT USABLE H2 WITHDRAWN TO END INTERVAL(I)
      C              * PCO2WD - PERCENT USABLE O2 WITHDRAWN TO END INTERVAL(I)
      C              * RHOCH2 - DENSITY OF H2 IN STORAGE TANK
      C              * RHOCO2 - DENSITY OF O2 IN STORAGE TANK
      C              * DQDWTH - DQ/DW * WTH(I)
      C              * DQDWTO - DQ/DW * WTO(I)
      C          *****
      C
        END
```

```
CCNFIG* PROC
C
      PARAMETER ICNF=100
C
      INTEGER CFUNCT,CFTYPE,CMTYPE,CITYPE,CNOPER,CNSTBY
C
      REAL LOD, ITHICK, MACH, NBAR
C
      COMMON /CCNFIG/ ICNFIG(6),CONFIG(ICNF,7),PRES(ICNF),TEMP(ICNF),
     1 WDOTN(ICNF),MACH(ICNF),MFLG(ICNF),WEIGHT(ICNF),WI(ICNF)
     2 ,ISTRT(2),KENDC(2),LCNFI(4,6),WTOFSY(5)
C
      DIMENSION FRCOEF(ICNF), LOD(ICNF), DIAM(ICNF), ITHICK(ICNF),
     1           NBAR  (ICNF), CODE(ICNF)
C
      EQUIVALENCE (FRCOEF(1),CONFIG(1,2)),(LOD    (1),CONFIG(1,3)),
     1            (DIAM   (1),CONFIG(1,4)),(ITHICK(1),CONFIG(1,5)),
     2            (NBAR   (1),CONFIG(1,6)),(CODE   (1),CONFIG(1,7))
C
      EQUIVALENCE       (CFUNCT,ICNFIG(1)),(CFTYPE,ICNFIG(2)),
     1(CMTYPE,ICNFIG(3)),(CITYPE,ICNFIG(4)),(CNOPER,ICNFIG(5)),
     2(CNSTBY,ICNFIG(6))
C
      EQUIVALENCE (ISTRT,IOSTT),(ISTRT(2),IHSTT),(KENDC,KOEND),
     2            (KENDC(2),KHEND)
C
      EQUIVALENCE       (WTOFSY    ,O2SWT),(WTOFSY(2),O2IWT),
     1 (WTOFSY(3),H2SWT),(WTOFSY(4),H2IWT),(WTOFSY(5),TTLSWT)
C
C
C        ***** CCNFIG VARIABLE DEFINITIONS.
C            *
C            * CONFIGURATION FUNCTION CODE AND TYPE.
C            * CFUNCT = 1, GAS          CFTYPE - 1=OXYGEN      2=HYDROGEN
C            *
C            *        = 2, ENGINE                1=HI-PRES.    2=LO-PRES.
C            *
C            *        = 3, LINE         CFTYPE = 10 A FIXED NUMBER
C            *
C            *        = 4, CONTROL      USES TWO DIGIT INDEX AS FOLLOWS,
C            *                          IDV = TENS DIGIT (10,20,ETC.)
C            *                          CFTYPE = UNITS DIGIT (1,2,ETC.)
C            *                          IDV = 10 FOR LIGHT WGT.CONTROL
C            *                              = 20 FOR MED. WGT.CONTROL.
C            *                              = 30 FOR HEAVY WGT.CONTROL
C            *                              = 40 FOR EXT.HEAVY CONTROL
C            *                          CFTYPE = 1 FOR VALVE
C            *                                 = 2 FOR REGULATOR
C            *                                 = 3 FOR ORIFICE
C            *                                 = 4 FOR FLOW METER
C            *
C            *        = 5, FITTING      USES TWO DIGIT INDEX AS FOLLOWS,
C            *                          LDV = TENS DIGIT (10,20,ETC.)
C            *                          CFTYPE = UNITS DIGIT (1,2,ETC.)
C            *                          LDV = 10 FOR USE IN LINE ONLY
C            *                                20 FOR 4-WAY TEE
C            *                                30 FOR 3-WAY TEE
```

```
********      CONFIG     ********

C         *                              40 FOR 90 DEG.ELBOW
C         *                              50 FOR 45 DEG.ELBOW
C         *                       CFTYPE = 1 FOR TEE
C         *                              = 2 FOR ELBOW
C         *
C         *       = 6, TAP        USES TWO DIGIT INDEX AS FOLLOWS,
C         *                       LDV = TENS DIGIT (10,20,ETC.)
C         *                       CFTYPE = UNITS DIGIT (1,2,ETC.)
C         *                       LDV = 10 FOR USE IN LINE ONLY
C         *                             20 FOR 4-WAY TEE
C         *                             30 FOR 3-WAY TEE
C         *                       CFTYPE = 1 FOR TEE
C         *
C         *       = 7, ACCUM                  NO OPTIONS
C         *
C         *       = 8, TANK                ( SEE TANK ROUTINE)
C         *
C         *       = 9, PUMP       USES TWO DIGIT INDEX AS FOLLOWS,
C         *                       JOPTN = TENS DIGIT (10,20,ETC.)
C         *                       CFTYPE = UNITS DIGIT (1,2,ETC.)
C         *                       JOPTN = 10 FOR MIN. POWER PUMP
C         *                       JOPTN = 20 FOR MIN. WGT. PUMP
C         *                       CFTYPE = 1 FOR HI-PRESS. PUMP
C         *                       CFTYPE = 2 FOR LO-PRESS XFER PUMP
C         *
C         *       = 10, HEX                1=HI-PRES.   2=LO-PRES.
C         *
C         *       = 11, END
C         *
C         * CMTYPE - CONFIGURATION MATERIAL TYPE.
C         *          CMTYPE = 1, 321/347 STAINLESS STEEL
C         *                 = 2, 2219-T87 ALUMINUM ALLOY
C         *                 = 3, 6061-T6  ALUMINUM ALLOY
C         *                 = 4, INCONEL-718 ALLOY
C         *                 = 5, TITANIUM TI-6AL-4V ALLOY
C         *                 = 6, CRES VACUUM JACKETED LINE
C         *                 = 7, 2219 VACUUM JACKETED LINE
C         *
C         * CITYPE - CONFIGURATION INSULATION TYPE.
C         *          CITYPE = 1, DBL.ALUM.MYLAR/SILK NET
C         *                   2, DBL.GOLD.MYLAR/SILK NET
C         *                   3, DBL.ALUM.MYLAR/TISSUE GLASS
C         *                   4, CRINK.DBL.ALUM. MYLAR
C         *                   5, NRC-2 CRINKLED ALUMINIZED MYLAR
C         *                   6, SUPERFLOC
C         *                   7, MICROSPHERES (104-135 MICRON)
C         *                   8, POLYURETHANE FOAM
C         *                   9, FIBERGLASS BATTING (JM)
C         *
C         * CNOPER - NUMBER OF OPERATIONAL UNITS (CFUNCT)
C         *
C         * CNSTBY - NUMBER OF STANDBY UNITS (CFUNCT)
C         *
C         * CONFIG - CONFIGURATION TABLE
C         *          COLUMN 1 CONTAINS THE ABOVE 6 VARIABLES PACKED
C         *                   ONE PER BYTE IN THE ORDER THEY ARE
C         *                   LISTED FROM LEFT TO RIGHT IN THE WORD.
C         *          COLUMN 2 CONTAINS THE FLOW FRICTION COEFICIENT.
```

```
********   CCNFIG   ********
      C          *          COLUMN 3 CONTAINS THE LENGTH OF A LINE OR THE
      C          *                   EFFECTIVE L/D FOR OTHER COMPONENTS.
      C          *          COLUMN 4 CONTAINS THE DIAMETER OF A LINE.
      C          *          COLUMN 5 CONTAINS THE INSULATION THICKNESS FOR
      C          *                   A LINE.
      C          *
      C          * PRES   - PRESSURE AT EACH POINT IN THE CONFIGURATION.
      C          ******
      C
            END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - CCNTRL

```
CCNTRL* PROC
C
        PARAMETER NBRSR=9,NBRSY=5
C
        INTEGER  SCRIT,SYSNUM
C
        COMMON /CCNTRL/ INBLK(NBRSY,5,2),NAMSYS(NBRSY),SCRIT,SYSNUM
       1 ,INTGSY,MDTRC(11),KSUBC(NBRSY,NBRSR),LREPT,JAPUS(2,2)
C
C                       INBLK   -  CONTROLS INPUT SELECTION IN COMPIL
C                       SCRIT   =  1  FOR SUB-CRITICAL
C                               =  2  FOR SUPER-CRITICAL
C                       SYSNUM  =  1  ACPS
C                               =  2  APU
C                               =  3  EC/LSS
C                               =  4  FUEL CELL
C                               =  5  OMS
C
C      CARD COL.        MDTRC( ) - DIAGNOSTIC TRACE SWITCH FOR CRYCON (OFF=0)
C         (70)               (1) = 1  TURN ON ACCRES
C         (71)               (2) = 1  TURN ON ACQWT
C         (72)               (3) = 1  TURN ON APUSUB OR APUSUP
C         (73)               (4) = 1  TURN ON CMPCAL
C         (74)               (5) = 1  TURN ON FUELCL
C         (75)               (6) = 1  TURN ON CONSUM
C         (76)               (7) = 1  TURN ON ECLSS
C         (77)               (8) = 1  TURN ON LIQRES
C         (78)               (9) = 1  TURN ON TANK
C         (79)              (10) = 1  TURN ON TSIZEI
C         (80)              (11) = 1  TURN ON WTACC
C
C                       MDTRC(1) IS CARD COL 70,--- MDTRC(11) IS CARD COL 80
C                         OF THE SYSTEM SPECIFICATION CARD
C
C
  END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR - CDCYCL

```
CDCYCL* PROC
C
      PARAMETER  ICDL=30,ICDL1=2*ICDL,ICDL2=ICDL1+1
C
      COMMON /CDCYCL/ DCYCLT, NDCYCL, DCYCLE(ICDL1), NEOP(ICDL),
     1                PSI(ICDL), HP(ICDL), PAMB(ICDL), KCYCLE
     2               ,PKW(ICDL),RPRTIM(ICDL)
C
C         ***** CDCYCL VARIABLE DEFINITION
C             *
C             * PSI    - % OF ENGINE DEGRADATION DUE TO MIB (INPUT).
C             *
C             * NDCYCL - NUMBER OF ELEMENTS IN DCYCLE (INPUT).
C             *
C             * DCYCLE - ARRAY OF DUTY CYCLE TIMES. ODD SUBSCRIPTED
C             *          WORDS CONTAIN BURN TIMES, EVEN SUBSCRIPTED
C             *          WORDS CONTAIN COAST TIMES (INPUT).
C             *
C             * DCYCLT - TOTAL OPERATING TIME FROM DUTY CYCLE.
C             *
C             * NEOP   - NUMBER OF OPERATING UNITS.
C             *
C             * HP     - HORSEPOWER LOAD PER OPERATING UNIT.
C             *
C             * PAMB   - AMBIENT PRESSURE AROUND OPER. UNITS.
C             *
C             * KCYCLE - INTEGER COUNT OF (ICDL) INPUT VALUES
C             *
C             * PKW    - POWER IN KILOWATTS
C             *
C             * RPRTIM - CABIN OR AIRLOCK REPRESSURIZATION TIME
C             *
C         *****
C
       END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - CECLSS

```
CECLSS* PROC
C
      PARAMETER LX2=2, LXV=12
C
      REAL NIENTH
C
      REAL  N2LCON, N2RES, N2REPR, N2CONS, N2TOTU, N2LWT, N2RHO, N2H,
     1                    N2TEMP, N2LRES, LINDIA, NRHO, N2TEM
C
      COMMON /CILSS/ MDAYS, NCREW ,NRPRES, NDARES , O2FNOM, GLKRAT,
     1                TENVR, CABVOL,O2MCON, O2LCON , N2LCON, O2RES ,
     2                N2RES, POPO2 ,POPN2 , PVPO2  , PVPN2 , TO2IN ,
     3                TN2IN, GASWGT,O2REPR, N2REPR , O2CONS, N2CONS,
     4                HLSO , O2TOTU,N2TOTU, WDTMXO , WDTMXN, TEMPO2,
     5                HLSN , TEMPN2,TKOMXT, TKNMXT , ZFO   , ZFN    ,
     6                WVO2 , WVN2  ,OLKOTK, QLKNTK , WTVJO , WTVJN,
     7                NPHO  ,O2LRES,N2LRES, HWTOMX, HWTNMX,TWTOMX,
     8                PSET1 ,PSET2 ,TWTNMX, TOTWMX, TOTWAT, TOTPOW,
     9                N2TEM
C
      COMMON /CVLSS/ TLSNOM(LX2), RHOBEG(LX2), TKFTEM(LX2), TKFPRS(LX2),
     1                TAU   (LXV), WTO2  (LXV), WTN2  (LXV), O2MWT (LXV),
     2                O2LWT (LXV), N2LWT (LXV), WDOTOM(LXV), WDOTNN(LXV),
     3                WDOTOR(LXV), WDOTNR(LXV), WDTO2 (LXV), WDTN2 (LXV),
     4                TKO2DP(LXV), TKN2DP(LXV), PCOXWD(LXV), PCN2WD(LXV),
     5                O2RHO (LXV), N2RHO (LXV), O2TEMP(LXV), N2TEMP(LXV),
     6                DODMO2(LXV), DPDUO2(LXV), DODMN2(LXV), DPDUN2(LXV),
     7                O2H   (LXV), N2H   (LXV), QDTOR (LXV), QDTNR (LXV),
     8                HWATO2(LXV), HWATN2(LXV), QDTTKO(LXV), QDTTKN(LXV),
     9                TWATO2(LXV), TWATN2(LXV), RHOEND(LX2), WTRSID(LX2),
     T                VOLTK (LX2), ARETK (LX2), QO2LK (LXV), QN2LK (LXV),
     A                QREQDO(LXV), QREQDN(LXV), WTVNTO(LXV), WTVNTN(LXV),
     B                TOTWTL(LX2), DITK  (LX2), DIVJ  (LX2), ROFTU (LX2),
     C                WTPV  (LX2), HTRFLX(LX2), LINDIA(LX2), PLSNOM(LX2),
     D                HTRDIA(LX2), HTRLNG(LX2), TIM   (LXV), TNONOP(LXV)
C
C         *****
C          * MDAYS  - DURATION OF MISSION IN DAYS
C          * NCREW  - NUMBER OF PERSONEL IN CREW
C          * NRPRES - NUMBER OF AIRLOCK REPRESSURATIONS
C          * NDARES - NUMBER OF DAYS SUPPLY OF RESERVE O2 AND N2
C          * O2FNOM - OXYGEN REQUIRED FOR CREW - LBS PER MAN-DAY
C          * GLKRAT - VEHICLE GAS LEAKAGE RATE - LBS PER DAY (AS AIR)
C          * TLSNOM - NOMINAL O2 OR N2 DELIVERED TEMPERATURE
C          * RHOBEG - O2 OR N2 INITIAL LOADED DENSITY
C          * TKFTEM - O2 OR N2 FINAL TANK TEMPERATURE
C          * TKFPRS - O2 OR N2 FINAL TANK PRESSURE
C          * TENVR  - LSS ENVIRONMENTAL TEMPERATURE (AMBIENT)
C          * CABVOL - CABIN OR AIRLOCK VOLUME - CU.FT.
C          *
C         *****
  END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR - CENG

```
CENG*   PROC
C
        PARAMETER  NECDL = 30
C
        REAL MIXRAT,ISP
C
        COMMON /CENG/GITEMP,GIPRES,THRUST,PSUBC ,EXPRAT,MIXRAT,NENG,
       1             ISP   ,WDOT  ,TIPWT ,ENGWT ,WDOTI(2),WDOTT(2),
       2             WDOTJ(NECDL,2)
C
C           ***** CENG VARIABLE DEFINITION
C               *
C               * GITEMP - GAS INLET TEMPERATURE (INPUT).
C               *
C               * GIPRES - GAS INLET PRESSURE (INPUT).
C               *
C               * THRUST - ENGINE THRUST (INPUT).
C               *
C               * PSUBC  - CHAMBER PRESURE (INPUT).
C               *
C               * EXPRAT - EXPANSION RATIO (INPUT).
C               *
C               * MIXRAT - MIXTURE RATIO (INPUT).
C               *
C               * NENG   - NUMBER OF ENGINES.
C               *
C               * ISP    - SPECIFIC IMPULSE (CALC IN ENGINE).
C               *
C               * WDOT   - FLOW RATE (CALC IN ENGINE).
C               *
C               * TIPWT  - TOTAL IMPULSE PROPELLANT WEIGHT (CALC IN ENGINE).
C               *
C               * ENGWT  - ENGINE WEIGHT (CALC IN ENGINE).
C               *
C               * WDOTI  - FLOW RATE FOR O2 AND H2 FROM ENGINE TO TAP.
C               *
C               * WDOTT  - FLOW RATE FOR O2 AND H2 FROM TAP THROUGH REST
C           *****          OF SYSTEM.
C
  END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR – CFLRAT

```
CFLRAT* PROC
C
      COMMON /CFLRAT/ EQRT(6,2),LFRT(3,7)
C
      EQUIVALENCE          (EQRT       ,WFL05 ),(EQRT(1,2),WFL07 ),
     1 (EQRT(2,1),WFL06 ),(EQRT(2,2),WFL08 ),(EQRT(3,1),WDTPTO),
     2 (EQRT(3,2),WDTPTF),(EQRT(4,1),WFL01 ),(EQRT(4,2),WFL03 ),
     3 (EQRT(5,1),WFL02 ),(EQRT(5,2),WFL04 ),(EQRT(6,1),WDHXTO),
     4 (EQRT(6,2),WDHXTF)
C
 END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - CFUID

```
CFLUID* PROC
C
C
 END
```

******** FUNCTIØN CFTW

```
      FUNCTION CFTW (D,P,IDV)
C
      REAL K1,K2,K3,K4
C
      DIMENSION K1(4),K2(4),K3(4),K4(4),C1(4),C2(4),C3(4),C4(4)
C
      DATA K1/0.040,0.057,0.073,0.090/
      DATA K2/0.057,0.073,0.090,0.107/
      DATA K3/1.000,2.500,3.300,5.500/
      DATA K4/2.500,3.300,5.500,7.700/
      DATA C1/1.750,3.950,5.730,8.910/
      DATA C2/3.950,5.730,8.910,12.35/
      DATA C3/0.800,1.500,2.500,3.500/
      DATA C4/1.500,2.500,3.500,4.500/
C
C                     SET IDV TO EXTRA HEAVY IF NOT INPUT
      IF (IDV .EQ. 0) IDV = 4
      IF(D.LE.1.) GO TO 10
      IF(P.GT.300.0.AND.D.GT.3.5) GO TO 5
      IF(P.GT.1000.0.AND.D.LE.3.5) GO TO 5
      CFTW = K1(IDV)*D*D*D + C1(IDV)
      RETURN
    5 CFTW = K2(IDV)*D*D*D + C2(IDV)
      RETURN
   10 IF(P.GE.1000.) GO TO 15
      CFTW = K3(IDV)*D + C3(IDV)
      RETURN
   15 CFTW = K4(IDV)*D + C4(IDV)
      RETURN
      END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR - CFUEL

```
CFUEL* PROC
C
      PARAMETER LFC = 12, LFD = 2
C
      REAL MRFC
C
      COMMON /CIFUEL/ SRCFC , MRFC  , POWTOT, WRFORP , WOCONS, WHCONS,
     1                PKWMAX, QFCTOT, QDTFC , TF2IIN , TF2IOU, TFOFC ,
     2                TFHFC , PFOFC , PFHFC , QTOTR  , QEXCES, WF2IMX,
     3                DQAMIN, TKOMAX, TKHMAX, QMXTKO , QMXTKH, WDTCFO,
     4                WDTCFH, WOCMP , WHCMP , POWMAX , DELTCP, WRRSRV,
     5                WORSRV, WHRSRV, QLEAKO, QLEAKH , WVHO  , WVHH  ,
     6                WOVENT, WHVENT, SPWT1 , SPWT2 , NFCOP , NFCSTB ,
     7                SPWTFC, FCWGT , PRFCOP, WDTFMX, PLSET1, PLSET2,
     8                FCVOLT
C
      COMMON /CVFUEL/ WRP   (LFC),WDTFCO(LFC),WDTFCH(LFC),WDOTMX(LFD),
     1                QAVAIL(LFC),WDTF2I(LFC),PRFCMN(LFD),TFCNOM(LFD),
     2                TKO2WD(LFC),TKH2WD(LFC),PCWDO2(LFC),PCWDH2(LFC),
     3                RHOTO2(LFC),RHOTH2(LFC),DODWO (LFC),DODWH (LFC),
     4                TKO   (LFC),TKH   (LFC),HTKO  (LFC),HTKH  (LFC),
     5                Q1ODTR(LFC),Q1HDTR(LFC),WDT1FO(LFC),WDT1FH(LFC),
     6                Q2ODTR(LFC),Q2HDTR(LFC),WDT2FO(LFC),WDT2FH(LFC),
     7                CSRVFO(LFC),CSRVFH(LFC),PHIFO2(LFC),PHIFH2(LFC),
     8                QSUMR (LFC),DQANET(LFC),RHOFIL(LFD),RHOFIN(LFD),
     9                WTRES (LFD),VOLTNK(LFD),AREATK(LFD),QLKO  (LFC),
     T                QLKH (LFC),WRTOTL(LFD), DIATK (LFD),DIAVJ (LFD),
     1                WCIRCP(LFD),RHOFTU(LFD),WTPVT (LFD),WTVJ  (LFD),
     B                WORFP (LFC),WHRFP (LFC),VJANUL(LFD),TKMXDI(LFD),
     C                PRGRAT(LFD),PRGTIM(LFD),PRGINT(LFD),PURGAS(LFD)
C
C          *****
C              *
C              *
C          *****
C
 END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR – CHEX

```
CHEX*  PROC
C
       PARAMETER  MHX=10, MXH2 = 2*MHX+1, MXH3 = 4*MHX+1, MXH4 = 6*MHX+1,
      1    MXH5 = 8*MHX+1, MXH6 =10*MHX+1, MXH7 =12*MHX+1, MXH8 =14*MHX+1,
      2    MXH9 =16*MHX+1, MXH10=18*MHX+1, MXH11=20*MHX+1, MXH0 =22*MHX,
      3    MYH2 = 6*MHX+1, MYH3 =12*MHX+1, MYH4 =18*MHX+1, MYH5 =24*MHX+1,
      4    MYH6 =30*MHX+1, MYH7 =36*MHX+1, MYH8 =42*MHX+1, MYH0 =48*MHX
C
       REAL NTUU
C
       COMMON /CHEX/ NUMHEX, HEXCIT(MHX,2), HEXCOT(MHX,2), CPCLDF(MHX,2),
      1        WDOTCF(MHX,2), HEXHIT(MHX,2), HEXHOT(MHX,2), CPHOTF(MHX,2),
      2        WDOTH (MHX,2), EPSLNC(MHX,2), EPSLNH(MHX,2), EPSLNS(MHX,2),
      3        HEXHIP(MHX,2), HEXHOP(MHX,2), HEXCIP(MHX,2), HEXCOP(MHX,2),
      4        HXHDLP(MHX,2), HXCDLP(MHX,2), HXMRAT(MHX,2), HXASSY(MHX,2),
      5        WGGFU (MHX,2), WGGFX (MHX,2), WGGSBT(MHX,2), WHXTOT(MHX,2),
      6        IISU  (MHX,2), NSSK  (MHX,2), NSUK  (MHX,2), UCODE (MHX,2),
      7        TCRU  (MHX,3,2), FWDTHU(MHX,3,2), FDPCU (MHX,3,2),
      8        CRU   (MHX,3,2), NTUU  (MHX,3,2), UAU   (MHX,3,2),
      9        WOUAU (MHX,3,2), WTHXU (MHX,3,2), UNAM(2,4),WDHX(2),
      T        LHX1(4,10),LHX2(4,14),LHX3(5),UOA   (MHX,2), DH    (MHX,2),
      A        HXCODE(MHX,2), HLNGTH(MHX,2)
C
       DIMENSION UOTHX (MXH0), UOTHY (MYH0)
C
       EQUIVALENCE            (UOTHX        ,HEXCIT),(UOTHX(MXH2 ),HEXCOT),
      1(UOTHX(MXH3 ),CPCLDF),(UOTHX(MXH4 ),WDOTCF),(UOTHX(MXH5 ),HEXHIT),
      2(UOTHX(MXH6 ),HEXHOT),(UOTHX(MXH7 ),CPHOTF),(UOTHX(MXH8 ),WDOTH ),
      3(UOTHX(MXH9 ),EPSLNC),(UOTHX(MXH10),EPSLNH),(UOTHX(MXH11),EPSLNS),
      4(UOTHY        ,TCRU  ),(UOTHY(MYH2 ),FWDTHU),(UOTHY(MYH3 ),FDPCU ),
      5(UOTHY(MYH4 ),CRU   ),(UOTHY(MYH5 ),NTUU  ),(UOTHY(MYH6 ),UAU   ),
      6(UOTHY(MYH7 ),WOUAU ),(UOTHY(MYH8 ),WTHXU )
C
C          ***** HEXHIT - HEX HOT INLET  TEMPERATURE  (HEX GAS GEN, TC)
C              *
C              * HEXHOT - HEX HOT OUTLET TEMPERATURE
C              *
C              * HEXCIT - HEX COLD INLET TEMPERATURE
C              *
C              * HEXCOT - HEX COLD OUTLET TEMPERATURE
C              *
C              * HEXHIP - HEX HOT INLET  PRESSURE     (HEX GAS GEN, PC)
C              *
C              * HEXHOP - HEX HOT OUTLET PRESSURE
C              *
C              * HEXCIP - HEX COLD INLET PRESSURE
C              *
C              * HEXCOP - HEX COLD OUTLET PRESSURE
C              *
C              * HXHDLP - HEX HOT  SIDE DELTA-PRESSURE
C              *
C              * HXCDLP - HEX COLD SIDE DELTA-PRESSURE
C              *
C              * HXMRAT - MIXTURE RATIO (O/F) OF HEX GAS GENERATOR
C              *
C              * WDOTH  - HEX HOT GAS FLOW RATE
C              *
```

```
*********     CHEX     *********
      C            * WDHX    - FLORAT ESTIMATE OF HEX FLOW REQD (EACH FLUID)
      C            *
      C            * WHXTOT  - HEX UNIT WEIGHT
      C            *
      C            * HXASSY  - HEX ASSEMBLY WEIGHT (HEX + GAS GEN.)
      C            *
      C            * HXCODE  - HEX CODE SYMBOL NAME
      C            *
      C            * NOTE --- THE ABOVE VARIABLES ARE SPECIED FOR OXYGEN IN
      C            *          THE FIRST WORD AND HYDROGEN IN THE SECOND
      C            *          WORD IN THE SECOND SUBSCRIPT.
      C            *
      C       *****
       END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - CHSØRC

```
CHSORC* PROC
C
        PARAMETER  MHSC = 10
C
        INTEGER HSTYPE
C
        COMMON /CHSORC/ HSTYPE(MHSC,2), HSMRAT(MHSC,2), HSOTEM(MHSC,2),
       1 HSAEE(MHSC,2), HSPRES(MHSC,2), HSWGHT(MHSC,2), HSGTOT(MHSC,2),
       2 HSGCPE(MHSC,2), HSQREQ(MHSC,2), HFTOT (2),     QTREQ (2),
       3 NUMHSA, LHS1(5,6), LHS2(5,14), ELCPOW(MHSC,2)
C
C            ***** HSTYPE - TYPE OF HEAT SOURCE
C                *              = 1, GAS GENERATOR ONLY
C                *              = 2, WASTE HEAT INPUT ONLY
C                *              = 3, GAS GEN. AND WASTE HEAT INPUT
C                *
C                * HSMRAT - HEAT SOURCE MIXTURE RATIO
C                *
C                * HSOTEM - HEAT SOURCE OUTLET TEMPERATURE
C                *
C                * HSAEE  - HEAT SOURCE AVAILABLE ENERGY
C                *
C                * HSPRES - HEAT SOURCE OUTLET PRESSURE
C                *
C                * HSWGHT - HEAT SOURCE WEIGHT
C                *
C                * HSGTOT - TOTAL HOT FLUID WEIGHT
C                *
C                * HSGCPE - SPECIFIC HEAT OF HOT FLUID
C                *
C                * HSQREQ - TOTAL HEAT REQUIRED FROM HOT FLUID
C                *
C                * HFTOT  - CUMULATIVE HOT FLUID FOR SYSTEM
C                *
C                * QTREQ  - CUMULATIVE HEAT REQUIRED FOR SYSTEM
C                *
C                * NUMHSO - NUMBER OF HEAT SOURCE SETS
C                *
C                * LHS1   - HEAT SOURCE OUTPUT LABELS
C                *
C                * ELCPOW - ELECTRIC HEAT SOURCE - WATTS
C                *
C                * IN THE ABOVE VARIABLES THE FIRST WORD OF THE SECOND
C                * ARRAY IS FOR OXYGEN, AND THE SECOND WORD IS FOR
C                * HYDROGEN.
C            *****
C
 END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - CHTX

```
CHTX* PROC
C
C        *****
C             *
C             * THE VARIABLES IN CHTX  ARE FOR OUTPUT USE AND ARE
C             * DEFINED IN S.R. HEATEX
C             *
C             * NOTE --- THE ABOVE VARIABLES ARE SPECIED FOR OXYGEN IN
C             *          THE FIRST WORD AND HYDROGEN IN THE SECOND
C             *          WORD IN THE SECOND SUBSCRIPT.
C             *
C        *****
 END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR – CIØUNT

```
CIOUNT* PROC
C
       COMMON /CIOUNT/ IOUNIT(14),IIN,IOT
C
       EQUIVALENCE         (MURD1 ,IOUNIT( 1)),(MURD2 ,IOUNIT( 2)),
     1(MURD3 ,IOUNIT( 3)),(MURD4 ,IOUNIT( 4)),(MURD5 ,IOUNIT( 5)),
     2(MURD6 ,IOUNIT( 6)),(NTAPE1,IOUNIT( 7)),(NTAPE2,IOUNIT( 8)),
     3(NTAPE3,IOUNIT( 9)),(NTAPE4,IOUNIT(10)),(NTAPE5,IOUNIT(11)),
     4(NTAPE6,IOUNIT(12)),(NTAPE7,IOUNIT(13)),(NTAPE8,IOUNIT(14))
C
C
C           IOUNIT - AND ARRAY OF VALUES WITH ARE THE LOGICAL UNIT NUMBERS
C                    AS DETAILED IN SEC. 4.4.1 OF THE LMSC DIGITAL
C                    COMPUTER SYSTEMS MANUAL (LMSC-685147).
C
 END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR – CKEYS

```
CKEYS*  PROC
C
      COMMON /CKEYS/ KEY1,KEY2
C
 END
```

********   PRØCEDURE DEFINITIØN PRØCESSØR - CMATRL

```
CMATRL* PROC
C
      REAL MINTHK
C
          COMMON/CMATRL/RHOL(10),RHOI(10),RHOIS(10),MINTHK(15)
C
C
C     **** RHOL(I) IS THE DENSITY OF THE SELECTED LINE MATL.  (LB/CU.FT)
C
C        *  I = 1 ,FOR 321/347 STAINLESS STEEL
C        *  I = 2 ,FOR 2219-T87 ALUM. ALLOY
C        *  I = 3 ,FOR 6061-TL  ALUM. ALLOY
C        *  I = 4 ,FOR INCONEL-718
C        *  I = 5 ,FOR TITANIUM TI-6AL-4V ALLOY
C        *
C     ****
C
C
C        *
C     **** RHOI(I)-THE DENSITY OF THE CHOSEN INSULATION MATRL.(LB/CU.FT)
C        * OPTIMIZED LAYER DENSITY VALUES.
C        *
C        *  I = 1, FOR DOUBLE ALUMINIZED MYLAR/SILK NET (50 LAYER/INCH)
C        *  I = 2, FOR DOUBLE GOLDIZED MYLAR/SILK NET   ( 50 LAYER/INCH)
C        *  I = 3, FOR DOUBLE ALUMINIZED MYLAR/TISSUE GLASS (100-LAY/IN)
C        *  I = 4, FOR CRINKLED DBLE.ALUM.MYLAR/TISS.GLASS  ( 30-LAY/IN)
C        *  I = 5, FOR NRC-2 CRINKLED ALUM. MYLAR       ( 40 LAYER/INCH)
C        *  I = 6, FOR SUPERFLOC                        ( 30 LAYER/INCH)
C        *  I = 7, FOR MICROSPHERES (104 MICRON DIA.)   (PER CUBIC FOOT)
C        *  I = 8, FOR POLYURETHANE FOAM                (PER CUBIC FOOT)
C        *  I = 9, FOR FIBERGLASS (J.M)(800-1200 DEG.R) (PER CUBIC FOOT)
C        *
C        * REF.'HDBK FOR THERMAL DESIGN'-LMSC-A847882,VOL.2,25 JUNE 1967
C        *     'MICROSPHERES'-DATA OF PARMLEY AND CUNNINGHAM, LMSC.
C        *
C     ****
C
C
C     **** RHOIS(I)--DENSITY OF THE CHOSEN INSULATION MATRL.(LB/CU.FT)
C        * ON PER LAYER BASIS FOR SPECIFYING LAYER FENSITY.
C        *
C        *  I = 1, FOR DOUBLE ALUMINIZED MYLAR/SILK NET (PER LAYER/INCH)
C        *  I = 2, FOR DOUBLE GOLDIZED MYLAR/SILK NET   (PER LAYER/INCH)
C        *  I = 3, FOR DOUBLE ALUMINIZED MYLAR/TISSUE GLASS (PER-LAY/IN)
C        *  I = 4, FOR CRINKLED DBLE.ALUM.MYLAR/TISS.GLASS  (PER-LAY/IN)
C        *  I = 5, FOR NRC-2 CRINKLED ALUM. MYLAR       (PER LAYER/INCH)
C        *  I = 6, FOR SUPERFLOC                        (PER LAYER/INCH)
C        *  I = 7, FOR MICROSPHERES (104 MICRON DIA.)   (PER CUBIC FOOT)
C        *  I = 8, FOR POLYURETHANE FOAM                (PER CUBIC FOOT)
C        *  I = 9, FOR FIBERGLASS (J.M)(800-1200 DEG.R) (PER CUBIC FOOT)
C        * REF.'HDBK FOR THERMAL DESIGN'-LMSC-A847882,VOL.2,25 JUNE 1967
C        *     'MICROSPHERES'-DATA OF PARMLEY AND CUNNINGHAM, LMSC.
C        *
C     ****
C
C                 MINIMUM THICKNESS DATA  (INCHES)
C
```

```
*********     CMATRL     *********

       C
        END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR - CMØTØR

```
CMOTOR* PROC
C
       REAL MEFF,MSS
C
       COMMON /CMOTOR/MTYPE,MEFF,MSS,PDNSTY,BWEGHT(2)
C
 END
```

SUBRØUTINE CMPCAL ********

```
635717*TPFS.CMPCAL
C
C                     * * * * * * * * * * * * * * * * * * * * * * * * *
C                     * ROUTINE NAME - PRESSURE DROP COMPUTATION      *
C                     *                FOR ENTIRE CONFIGURATION       *
C                     * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                     * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                     * DATE CODED   - 3/17/70                       *
C                     * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE CMPCAL
C
      INTEGER GSTATE
C
      LOGICAL PAGE, JP
C
C
      REAL NPSPR
C
      INCLUDE CACCUM
      INCLUDE CCNFIG
      INCLUDE CCNTRL
      INCLUDE CDCYCL
      INCLUDE CENG
      INCLUDE CHEX
      INCLUDE CFUEL
      INCLUDE CHSORC
      INCLUDE CIOUNT
      INCLUDE CNAMES
      INCLUDE CMOTOR
      INCLUDE CONST
      INCLUDE CPAGE
      INCLUDE CPUMP
      INCLUDE CTANK
      INCLUDE CTURBN
      INCLUDE TABLOK
C
C         ***** INITIALIZE THE ROUTINE
C
      IDX = 0
      ISIGN = 1
      JKM = 0
      WGGTOT(1) = 0.0
      WGGTOT(2) = 0.
      HFTOT(1) = 0.0
      HFTOT(2) = 0.0
      QTREQ(1) = 0.0
      QTREQ(2) = 0.0
      C1 = 1152.0/(GRAVTY*PI**2)
      IF(PAGE(0)) WRITE (IOT,6050)
      WRITE (IOT,6020)
      JP = PAGE (3)
C
C         ***** START OF CONFIGURATION PROCESSING LOOP
C
      DO 1000 II=1,ICNF
      IDX = IDX + ISIGN
      MACH(IDX) = 0.0
```

```
      MFLG(IDX) = 6H
      CALL GETCON(IDX)
C
C          ***** BRANCH TO THE REQUIRED CONFIGURATION TYPE. SEE CCNFIG
C          ***** FOR BRANCH DEFINITIONS.
C
      GO TO (100,200,300,400,450,500,450,450,400,405,600,700,800,900,
     1 230,250,270,1100), CFUNCT
C
C          ***** SETUP THE GAS TYPE *****
C
  100 IGAS = CFTYPE
      GSTATE = ICNFIG(5)
      IF (IGAS .EQ. JKM) GO TO 110
      JKM = IGAS
      ISIGN = 1
      ISTRT(IGAS) = IDX + 1
      JX = 0
  110 CONTINUE
C
      IF(IGAS.EQ.2.AND.GSTATE.EQ.1) GO TO 111
      GO TO 112
  111 IF(PAGE(0)) WRITE (IOT,6051)
      WRITE (IOT,6020)
      JP = PAGE(3)
C
  112 CONTINUE
C
      IF(I1.EQ.1) GO TO 999
      IF(IGAS.EQ.2.AND.GSTATE.EQ.1) GO TO 999
      PRES(IDX) = PRES(IDX - ISIGN)
      WDOTN(IDX) = WDOTN(IDX-ISIGN)
      TEMP(IDX) = TEMP(IDX - ISIGN)
      GO TO 999
C
C          ***** PROCESS AN ENGINE *****
C
  200 WDOTN(IDX) = WDOTI(IGAS)
      PRES(IDX) = GIPRES
      TEMP(IDX) = GITEMP
      WEIGHT(IDX) = ENGWT
      GO TO 999
C
C          ***** PROCESS AN APU TURBINE UNIT *****
C
  230 WDOTN(IDX) = WDOTI(IGAS)
      PRES(IDX) = HEXCOP(1,IGAS)
      TEMP(IDX) = HEXCOT(1,IGAS)
      GO TO 999
C
C          ***** PROCESS A FUEL CELL *****
C
  250 WDOTN(IDX) = WDOTI(IGAS)
      PRES(IDX) = PRFCOP
      TEMP(IDX) = TFCNOM(IGAS)
      ISIGN  =  1
      GO TO 999
C
```

```
C
C          ***** PROCESS A LINE *****
C
  300 FLD = FRCOEF(IDX)*LOD(IDX)/DIAM(IDX)
      LDV = CFTYPE/10
      CFTYPE = CFTYPE - LDV * 10
  310 WDOTN(IDX) = WDOTN(IDX-ISIGN)
      TEMP(IDX) = TEMP(IDX-ISIGN)
      GO TO 510
C
C          ***** PROCESS A CONTROL *****
C
  400 FLD = FRCOEF(IDX)*LOD(IDX)
      IDV = CFTYPE /10
      CFTYPE = CFTYPE - IDV * 10
      DIAM(IDX) = AMIN1(DIAM(IDX+1),DIAM(IDX-1))
      GO TO 310
C
C          *****  PROCESS A REGULATOR  *****
C
  405 FLD = FRCOEF(IDX)*LOD(IDX)
      IDV = CFTYPE /10
      CFTYPE = CFTYPE - IDV * 10
      DIAM(IDX) = AMIN1(DIAM(IDX+1),DIAM(IDX-1))
      WDOTN(IDX) = WDOTN(IDX-ISIGN)
      TEMP(IDX) = TEMP(IDX-ISIGN)
C
      IX = IDX - ISIGN
      IF(APRES(IGAS) .EQ. 0.0) GO TO 406
      DLPREG = (APRES(IGAS) - ANDELP(IGAS)/2.0) - PRES(IX)
      PRES(IDX) = PRES(IX) + ISIGN * DLPREG
      GO TO 561
C
  406 CONTINUE
      DLPREG = HEXCOP(1,IGAS) - PRES(IX)
      PRES(IDX) = PRES(IX) + ISIGN * DLPREG
      GO TO 561
C
C          ***** PROCESS A FITTING *****
C
  450 FLD = FRCOEF(IDX) * LOD(IDX)
      LDV = CFTYPE/10
      CFTYPE = CFTYPE - LDV * 10
      DIAM(IDX) = AMIN1(DIAM(IDX+1),DIAM(IDX-1))
      GO TO 310
C
C          ***** PROCESS A TAP *****
C
  500 WDOTN(IDX) = WDOTT(IGAS)
      LDV = CFTYPE/10
      CFTYPE = CFTYPE - LDV * 10
      FLD = FRCOEF(IDX)*LOD(IDX)
      TEMP(IDX) = TEMP(IDX-ISIGN)
      DIAM(IDX) = AMIN1(DIAM(IDX+1),DIAM(IDX-1))
C
C          *****  COMPUTE LINE, CONTROL, FITTING OR TAP DELTA PRESSURE.
C
  510 IX = IDX - ISIGN
```

```
      GO TO(520,550),GSTATE
C
C         ***** DELTA PRESSURE WHEN GASEOUS
C
C                    CALC. RHO OF GAS
  520 CALL GSDNST (IGAS,TEMP(IX),PRES(IX),FHO)
      DELP = CI*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IDX)**4)
C
C         ***** IF PCT. OF PRESSURE CHANGE EXCEEDS ONE PCT. - RECOMPUTE
C         ***** DELTA-P, IF NOT, COMPUTE THE NEW PRESSURE
C
      IF(DELP/(PRES(IX) + DELP) - 0.01)560,560,530
C
C                    CALC. RHO OF GAS
  530 CALL GSDNST (IGAS,TEMP(IX),PRES(IX)+DELP/2.0,RHO)
      DELP = CI*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IDX)**4)
C
C         ***** AGAIN CHECK PCT. OF PRESSURE CHANGE. IF PCT. EXCEEDS
C               * 2.8 PCT. COMPUTE THE DELTA-P BY USE OF THE COMPRESSIBLE
C         ***** FLOW EQUATIONS. (REF.-RPL-TDR-64-25,VOL.I,REV.D)
C
      IF(DELP/(PRES(IX) + DELP) - 0.028)560,560,540
C
  540 A = PI*DIAM(IDX)**2/576.0
      CALL COMFLO(IDX,PRES(IX),TEMP(IX),FLD,A,WDOTN(IX)/CNOPER,IGAS,
     1 DELP)
      PRES(IDX) = PRES(IX) + ISIGN * DELP
      GO TO 561
C
C         ***** DELTA PRESSURE WHEN LIQUID
C
  550 CALL RHOLIQ(TEMP(IX),IGAS,RHO)
      DELP = CI*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IDX)**4)
C
C         ***** COMPUTE NEW PRESSURE
C
  560 PRES(IDX) = PRES(IX) + ISIGN*DELP
C
C         ***** COMPUTE THE GAS MACH NUMBER
C
      IF(GSTATE.EQ.2) GO TO 561
C
C                    CALC. RHO OF GAS
      CALL GSDNST (IGAS,TEMP(IX),PRES(IDX),RHO)
C
      IF(GSTATE.EQ.1) CALL VGVS(IDX,RHO,IGAS)
C
  561 CONTINUE
C
C         ***** COMPUTE LINE WEIGHT
C
      IF(CFUNCT.EQ.3) GO TO 562
      IF(CFUNCT.EQ.5) GO TO 562
      IF(CFUNCT.EQ.6) GO TO 562
      IF(CFUNCT.EQ.7) GO TO 562
      IF(CFUNCT.EQ.8) GO TO 562
      GO TO 570
C
```

```
*********     CMPCAL     *********

      562 CALL LWEGHT(IDX,LDV)
    C
          GO TO 999
    C
    C         ***** COMPUTE CONTROL, FITTING OR TAP WEIGHT
    C
      570 WEIGHT(IDX) = CFTW (DIAM(IDX),PRES(IDX),IDV)
          GO TO 999
    C
    C         ***** PROCESS AN ACCUMULATOR *****
    C
      600 PRES(IDX) = APRES(IGAS)
          TEMP(IDX) = TEMP(IDX - ISIGN)
          WDOTN(IDX) = WDOTN(IDX - ISIGN)
          INDXAC(IGAS) = IDX
          GO TO 999
    C
    C         ***** PROCESS A TANK OR SUPPLY *****
    C
      700 CFT = 1
          INDXTK(IGAS) = IDX
    C
          IF(SIPRES(IGAS,CFT)) 720,710,720
    C
    C         ***** IF NO TANK INPUT PRESSURE IS INPUT USE THE VALUE CALC.
    C
      710 SIPRES(IGAS,CFT) = PRES(IDX-ISIGN)
          GO TO 740
    C
    C         ***** CHECK THE TANK INPUT PRESSURE AGANIST THE REQUIRED
    C               * CALCULATED PRESURE. IF THE TANK INPUT PRESSURE IS LESS
    C               * THAN THE CALC. PRESSURE WRITE A DIAGNOSTIC MESSAGE AND
    C               * SET THE TANK INPUT PRESSURE = THE REQUIRED PRESSURE.
    C         ***** IF NOT CONTINUE WITH CALCULATIONS.
    C
      720 IF(SIPRES(IGAS,CFT) - PRES(IDX-ISIGN)) 730,740,740
    C
      730 WRITE (6,6000) SIPRES(IGAS,CFT), PRES(IDX-ISIGN)
          GO TO 710
    C
      740 PRES(IDX) = SIPRES(IGAS,CFT)
    C
    C         ***** DO THE SAME CHECKS FOR THE INPUT TANK TEMPRATURE.
    C
          IF(SITEMP(IGAS,CFT)) 760,750,760
      750 SITEMP(IGAS,CFT) = TEMP(IDX-ISIGN)
    C
      760 TEMP(IDX) = SITEMP(IGAS,CFT)
    C
          WDOTN(IDX) = WDOTN(IDX-ISIGN)
          GO TO 999
    C
    C         ***** PROCESS A PUMP *****
    C
    C         ***** CHECK ISIGN TO SEE IF THE CONDITIONS ON BOTH SIDES OF
    C         ***** THE PUMP HAVE BEEN CALCULATED.
    C
      800 IF(ISIGN.LT.0) GO TO 825
```

```
            ISIGN = -1
      C
      C         ***** SEARCH FORWARD IN LINE FOR A SOURCE TANK
      C
            DO 810 I2=IDX,ICNF
            CALL GETCON(I2)
            IF(CFUNCT.EQ.12) GO TO 820
        810 CONTINUE
      C
            WRITE(6,6010) IDX
            CALL EXIT
      C
      C         ***** WHEN A SOURCE TANK IS FOUND, SETUP PRESSURE, TEMPRATURE,
      C         ***** FLOW RATE AND FLAG TO CONTINUE THE CALCULATIONS.
      C
        820 WDOTN(I2) = WDOTN(IDX-1)
            IDT = I2 - 1
            IDX = I2
      C
            PRES(IDX) = SIPRES(IGAS,1)
            TEMP(IDX) = SITEMP(IGAS,1)
            GO TO 1000
      C
      C         ***** COMPUTE THE WEIGHT OF THE PUMP, TURBINE, PROPELLANT AND
      C         ***** MOTOR DEPENDING ON THE TYPE OF PUMP.
      C
      C         ***** CHECK CFTYPE FOR HIGH OR LOW PRESSURE PUMP
      C
        825 ISIGN = 1
            PRES(IDX) = PRES(IDX+1)
            TEMP(IDX) = TEMP(IDX+1)
            WDOTN(IDX) = WDOTN(IDX+1)
      C
            JOPTN = CFTYPE/10
            CFTYPE = CFTYPE - JOPTN * 10
      C
            IF(CFTYPE.EQ.2) GO TO 840
      C
      C         ***** PROCESS THE HIGH PRESSURE PUMP
      C
      C
      C         ***** COMPUTE THE PUMP OR TURBOPUMP WEIGHT
      C
            JJOPT(IGAS) =JOPTN
            PTEMP(IGAS) =TEMP(IDX)
            PPRES(IGAS) =PRES(IDX)
            PPDCH(IGAS) = PRES(IDX-1)
            PDELP = ABS(PRES(IDX-1) - PRES(IDX))
            PPDEL(IGAS) =PDELP
            PWDOT = WDOTN(IDX)/CNOPER
            PPWDT(IGAS) =PWDOT
            CALL RHOLIQ(TEMP(IDX),IGAS,RHO)
            PPRHO(IGAS) =RHO
      C                CALCULATE PUMP PARAMETERS
      C
            CALL PARPMP(IGAS,JOPTN,PDELP,PWDOT,PNPSH(IGAS),RHO,PMEF,V,
           1 E, WT, PNSG, NSTG, NPSPR)
            PMPEFF(IGAS) = PMEF
```

```
*********      CMPCAL     *********

            PMPVOL(IGAS) = V
            PMPOW(IGAS) =  E
            PSPD(IGAS) = PNSG
            PSTAGE(IGAS) = NSTG
            PNPSPR(IGAS) = NPSPR
            PWEGHT(IGAS) = WT * (CNOPER + CNSTBY)
C
C              ***** CHECK PTYPE FOR PUMP OR TURBOPUMP *****
C
            IF(PTYPE.EQ.1) GO TO 830
C
C              ***** COMPUTE THE TURBINE WEIGHT
C
            CALL TURBN(IGAS,TRBWGT)
            TWEGHT(IGAS) = TRBWGT * (CNOPER + CNSTBY)
C
C              ***** COMPUTE THE FLOWRATE OF THE GAS GENERATOR AND ITS WEIGHT
C
            TMEAN1 = (TITEMP(1) - TOTEMP(1))/2.0
            TMEAN2 = (TITEMP(2) - TOTEMP(2))/2.0
            CALL CSUBP1(TMEAN1,TMRATO(1),CPEP1)
            CALL CSUBP1(TMEAN2,TMRATO(2),CPEP2)
            DLHTP1 = CPEP1 * (TITEMP(1) - TOTEMP(1))
            DLHTP2 = CPEP2 * (TITEMP(2) - TOTEMP(2))
            CALL RHOLIQ(SITEMP(1,1),1,RHOLQ1)
            CALL RHOLIQ(SITEMP(2,1),2,RHOLQ2)
            BRAC1 = (0.185 * PPDEL(1))/(RHOLQ1 * PMPEFF(1) * TEFF(1) * DLHTP1)
            C1 = TMRATO(1)/(1.0 + TMRATO(1))
            C2 = 1.0/(1.0 + TMRATO(1))
            IF(PPDEL(2).EQ.0.0) PPDEL(2) = EPDELP(2)
            IF(PMPEFF(2).EQ.0.0) PMPEFF(2)= PEFF(2)
            BRAC2 = (0.185 * PPDEL(2))/(RHOLQ2 * PMPEFF(2) * TEFF(2) * DLHTP2)
            C3 = TMRATO(2)/(1.0 + TMRATO(2))
            C4 = 1.0/(1.0 + TMRATO(2))
            C5 = BRAC1 * C1
            C6 = BRAC1 * C2
            C7 = BRAC2 * C3
            C8 = BRAC2 * C4
            DI = (C5 + C8 + C6*C7 - C5*C8 - 1.0)
            WFLO1 = (WDOTI(1) * (C5*C8-C5) + WDOTI(2) * (-C5*C7))/DI * CNOPER
            WFLO2 = (WDOTI(1) * (-C6*C7) + WDOTI(2) * (C5*C7 - C7))/DI*CNOPER
            WFLO3 = (WDOTI(1) * (C6*C8 - C6) + WDOTI(2) * (-C6*C7))/DI*CNOPER
            WFLO4 = (WDOTI(1) * (-C6*C8) + WDOTI(2) * (C5*C8 - C8))/DI*CNOPER
            WDGGFR(1) = (WFLO1 + WFLO3) * CNOPER
            WDGGFR(2) = (WFLO2 + WFLO4) * CNOPER
C
            KK = 0
            WDOTG1 = 0.0
            WDOTG2 = 0.0
C
            DO 835 I2 = 1,NDCYCL,2
            KK = KK + 1
            WFLO5 = (WDOTJ(KK,1) * (C5*C8-C5) + WDOTJ(KK,2) * (-C5*C7))/DI
            WFLO6 = (WDOTJ(KK,1) * (-C6*C7) + WDOTJ(KK,2) * (C5*C7 -C7))/DI
            WFLO7 = (WDOTJ(KK,1) * (C6*C8 - C6) + WDOTJ(KK,2) * (-C6*C7))/DI
            WFLO8 = (WDOTJ(KK,1) * (-C6*C8) + WDOTJ(KK,2) * (C5*C8 -C8))/DI
            WFLO57 = (WFLO5 + WFLO7)/CNOPER
            WFLO68 = (WFLO6 + WFLO8)/CNOPER
```

```
      WDOTG1 = WDOTG1 + WFL057 * DCYCLE(I2)
      WDOTG2 = WDOTG2 + WFL068 * DCYCLE(I2)
  835 CONTINUE
C
      GWEGHT(1) = WDOTG1 * CNOPER
C
      GWEGHT(2) = WDOTG2 * CNOPER
C
C          ***** COMPUTE SYSTEM WEIGHT
C
      ATERM = 13.824204 - (0.01117823*TGGPC( IGAS   )) + (1.8632927E-5 *
     1(TGGPC( IGAS   )**2)) - (1.108423E-8 * (TGGPC( IGAS   )**3))
C
      BTERM = 7.9470262 - (.035636198*TGGPC( IGAS   )) + (6.4684644E-5 *
     1(TGGPC(IGAS    )**2)) - (3.7946E-8 * (TGGPC( IGAS   )**3))
C
      WTGGA = ATERM + BTERM * WDGGFR(IGAS)/CNOPER
      WGTGGA(IGAS) = WTGGA * (CNOPER + CNSTBY)
      WEIGHT(IDX) = PWEGHT(IGAS) + TWFGHT(IGAS) +WGTGGA(IGAS)
C
      GO TO 999
C
C
  830 CONTINUE
      WEIGHT(IDX) = PWEGHT(IGAS)
C
      GO TO 999
C
C          ***** PROCESS LOW PRESSURE SYSTEM
C          ***** FLUID TRANSFER PUMP
C
  840 CONTINUE
      IF((SYSNUM.EQ.2.OR.SYSNUM.EQ.4) .AND. (SCRIT.EQ.2)) GO TO 841
      CALL FINTAB (NTBID(15)+IGAS)
      XTAB(1) = TPEFF(IGAS)*100.0
      XTAB(2) = TPNPSH(IGAS)
      IF(TPDELP(IGAS))860,850,860
  850 TPDELP(IGAS) = PRES(IDX) - PRES(IDX-1)
  860 IF(TPWDOT(IGAS))880,870,880
  870 TPWDOT(IGAS) = WDOTN(IDX)/CNOPER
  880 XTAB(3) = TPDELP(IGAS)
      XTAB(4) = TPWDOT(IGAS)
      TPWGHT(IGAS) = MIPE(4,XTAB)*(CNOPER + CNSTBY)
C
C          ***** ELECTRIC MOTOR FOR TRANSFER PUMP
C
      CALL FINTAB (NTBID(16))
      CALL RHOLIQ(TEMP(IDX),IGAS,RHO)
      HP = (144.0*WDOTN(IDX)*(PRES(IDX) - PRES(IDX-1))/(550.0*PEFF*RHO*
     1     MEFF)
      XTAB(1) = HP
      XTAB(2) = MSS
      EMWGT = MIPE(2,XTAB) * (CNOPER + CNSTBY)
      WEIGHT(IDX) = EMWGT + TPWGHT(IGAS)
C
      KK = 0
      BWEGHT(IGAS) = 0.0
C
```

```
      DO 890 I2 = 1,NDCYCL,2
      KK = KK + 1
      HP = 144.0*WDOTJ(KK,IGAS)*(PRES(IDX) - PRES(IDX - 1))/(550.0*PEFF*
     1     RHO*MEFF)
      PB = HP*746.0*DCYCLE(I2)/3600.0
      BWEGHT(IGAS) = BWEGHT(IGAS) + PB/PDNSTY
  890 CONTINUE
      GO TO 999
C
  841 CONTINUE
      WEIGHT(IDX) = WCIRCP(IGAS)
      GO TO 999
C
C            ***** PROCESS A HEAT EXCHANGER *****
C
  900 IF(ISIGN.GT.0) GO TO 910
C
      WRITE (IOT,6005) ISIGN
C
  910 CONTINUE
      JX = JX + 1
      JHX = JX
      WDOTN(IDX) = WDOTN(IDX-ISIGN)
      WDOTCF(JX,IGAS) = WDOTN(IDX)
      UCODE(JX,IGAS) = CODE(IDX)
C
      HXDLP = HEXCIP(JX,IGAS) - HEXCOP(JX,IGAS)
C
      IF(HXDLP.GT.0.0) GO TO 911
      UCODE(JX,IGAS) = 6HNONE
      PRES(IDX) = PRES(IDX-1)
      TEMP(IDX) = TEMP(IDX-1)
      GO TO 1000
C
  911 CONTINUE
C
      HEXCOP(JX,IGAS) = PRES(IDX - 1)
      HEXCIP(JX,IGAS) = HEXCOP(JX,IGAS) + HXDLP
      IF (SCRIT .EQ. 1) GO TO 913
      IF (SYSNUM .EQ. 2 .OR. SYSNUM .EQ. 4) GO TO 912
  913 CONTINUE
C
C                    COMPUTE HEATEX PARAMETERS
C
      CALL HEATEX(IGAS,JHX,WDOTN(IDX),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS),
     1 HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
     2 HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOTH (JX,IGAS),
     3 WHXTOT(JX,IGAS))
C
C
C            ***** COMPUTE THE GAS GENERATOR ASSEMBLY WEIGHT *****
C
      CALL GASGEN(JX,IGAS)
C
  912 CONTINUE
C
      TEMP(IDX) = HEXCIT(JX,IGAS)
      DLPRES = HEXCIP(JX,IGAS) - HEXCOP(JX,IGAS)
```

```
      PRES(IDX)= PRES(IDX-ISIGN) + DLPRES*ISIGN
      WEIGHT(IDX) = WHXTOT(JX,IGAS)
C
C        ***** END OF CONFIGURATION PROCESSING LOOP *****
C
  999 CONTINUE
      IF (.NOT. PAGE(1)) GO TO 1998
C
C     *** PAGE HEADER HAS BEEN MOVED TO STATEMENT GROUP 100
C
 1998 CONTINUE
      KFUNCT = FNAME(CFUNCT)
      WRITE(IOT,6030)KFUNCT,CODE(IDX),CFTYPE,CNOPER,CNSTBY,ISIGN,IDX,
     1               IGAS,GSTATE,FRCOEF(IDX),LOD(IDX),DIAM(IDX),
     2               ITHICK(IDX),PRES(IDX),TEMP(IDX),WDOTN(IDX),
     3               WEIGHT(IDX),MACH(IDX),MFLG(IDX)
      IF (PRES(IDX) .GE. 0. .AND. TEMP(IDX) .GE. 0.) GO TO 998
      WRITE (IOT,6040)
      CALL EXIT
  998 IF (CFUNCT .EQ. 13) IDX = IDT
C
  270 CONTINUE
 1000 CONTINUE
 1100 CONTINUE
C                    USED BY WEIGHT SUMMARY OUTPUT
      KHEND = IDX - 1
      KOEND = IHSTT - 2
C
      IF((SYSNUM.EQ.2.OR.SYSNUM.EQ.4) .AND. (SCRIT.EQ.2)) GO TO 1150
C
C     OUTPUT THE HEAT EXCHANGER AND GAS GENERATOR DATA      *****
C
      CALL OTPHEX
C
C     OUTPUT THE PUMP AND TURBINE DATA         *****
C
      CALL OTPPMP
C
      CALL OTPTRB
C
 1150 CONTINUE
C
      RETURN
C
C        ***** OUTPUT FORMATS
C
 6000 FORMAT('0 *DIAGNOSTIC* TANK INPUT PRESSURE IS LESS THAN THE REQUIR
     1ED PRESSURE, TANK PRESSURE SET = REQUIRED PRESSURE,'/15X,'TANK INP
     2UT PRESSURE = ',F7.2,', REQUIRED PRESSURE = ',F7.2)
C
 6005 FORMAT('0 **ERROR** ISIGN =',I3,'THERE IS A CONFIGURATION ERROR'/)
C
 6010 FORMAT('0 *ERROR* A PUMP WAS ENCOUNTERED BUT NO TANK CAN BE FOUND.
     1 PUMP CONFIGURATION INDEX NUMBER = ',I3)
C
 6020 FORMAT('0  F   CODE   FT  NO  NS  IS IDX   G  GS   FCOEF      L/D
     1         DIAM  ITHICK    PRES      TEMP     WDOT  WEIGHT    MACH    M
     2FLAG'/' ')
```

```
C
 6030 FORMAT(2XA3,2XA6,I3,6I4,F9.6,F12.4,2F8.4,2F9.2,F7.2,F9.3,F10.7,
     13X,A6)
C
 6040 FORMAT (T44, '***  TERMINATE - NEGATIVE TEMP. OR PRES.  ***')
C
 6050 FORMAT(/T38,'*** SUMMARY OF COMPUTED SYSTEM CONFIGURATION PARAMETE
     1RS ***')
C
 6051 FORMAT(/T32,'*** SUMMARY OF COMPUTED SYSTEM CONFIGURATION PARAMETE
     1RS - CONTD. ***')
C
      END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - CNAMES

```
CNAMES* PROC
C
      INTEGER FNAME
C
      COMMON /CNAMES/ FNAME(18), L0(9,15), L1(21,2), L2(3,7), L3(4,3),
     1 L4(4,11), L5(4,5), L6(4,4), L7(4,23), L8(4,3), L9(4,5), L10(4,5),
     2 L11(4,20), L12(4,28), L13(4,28), JFLUID(2,3), KFLUID(2,2)
C
C          NAMES ARE GIVEN IN S.R. STODTA
C
      END
```

******** SUBRØUTINE CØMFLØ

```
C                   * * * * * * * * * * * * ** * * * * * * * * * * *
C                   * ROUTINE NAME - COMPRESSIBLE FLOW EQUATIONS *
C                   *                EVALUATION ROUTINE          *
C                   * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                   * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                   * DATE CODED   - 3/10/70                     *
C                   * * * * * * * * * * * * ** * * * * * * * * * * *
C
      SUBROUTINE COMFLO(IDX,P,T,FLD,A,WDOT,N,DELP)
C
C         ***** EXPLANATION OF THE CALLING SEQUENCE
C             *
C             * P    - PRESSURE
C             *
C             * T    - TEMPRATURE
C             *
C             * FLD  - RESISTANCE COEFICIENT
C             *
C             * A    - AREA
C             *
C             * WDOT - FLOW RATE
C             *
C             * N    - GAS TYPE NUMBER
C                      N = 1 FOR OXYGEN
C                      N = 2 FOR HYDROGEN
C             *          17 FOR HE
C             *
C             ***** DELP - COMPUTED PRESSURE DROP
C
      LOGICAL DIAG
C
      REAL M1, M2, M1MIN, M1MAX
C
      INCLUDE CCNFIG
      INCLUDE CIOUNT
      INCLUDE CONST
C
      DATA JBLNK,JAST1,JAST6/'      ','*     ','******'/
C
      MFLG(IDX) = JBLNK
C
C         ***** COMPUTE GAMMA = SPECIFIC HEAT AT CONSTANT PRESSURE/
C             *                 SPECIFIC HEAT AT CONSTANT VOLUME
C         *****                 FOR A SPECIFIC GAS
C
      IF (DIAG(0,6HCOMFLO)) WRITE (IOT,6010) N,P,T,FLD,A,WDOT,DELP
      IF(N.EQ.2) GO TO 60
      CALL CSUBP(T,P,N,CP)
      CP1 = CP
      CV1 = CSUBV(T,P,N)
      GMA = CP1/CV1
      GO TO 65
C
   60 GMA = HPTGAM(P,T)
C
   65 CONTINUE
C
```

```
C          ***** COMPUTE MACH NUMBER AT PRIOR CONDITIONS
C
      M2 = WDOT * SQRT(FINDR(N)*T/(GMA*GRAVTY))/(P*A*144.0)
C
C          ***** COMPUTE RESISTANCE COEFICIENT AT PRIOR CONDITIONS
C
      FLDMAX = FLODEQ(M2,GMA)
      DF = FLDMAX + FLD
C
C          ***** ITTERATE TO FIND MACH NUMBER AT CURENT CONDITIONS
C
      M1MIN = 1.E-10
      M1MAX = 1.0
C
      DO 40 I1 = 1,20
      M1 = (M1MAX + M1MIN)/2.0
      FAC = FLODEQ(M1,GMA) - DF
      IF(ABS(FAC) - 0.00005) 50,50,10
   10 IF(FAC) 20,50,30
   20 M1MAX = M1
      GO TO 40
   30 M1MIN = M1
   40 CONTINUE
C
      WRITE (6,6000) M1
C
C          ***** COMPUTE THE PRESSURE DROP
C
   50 A1 = M1**2
      A2 = M2**2
      P2OP1 = (M1/M2) * (2.0 + (GMA - 1.0) * A1)/(2.0 + (GMA - 1.0)* A2)
      DELP = P * (1.0/P2OP1 - 1.0)
C
C          ***** SET FINAL MACH NUMBER FROM ITERATION
C
      MACH(IDX) = M1
C
C          ***** CHECK MACH NUMBER, FLAG MACH NUMBER GREATER THAN 0.3
C                * WITH ONE ASTERISK, FLAG MACH NUMBER GREATER THAN
C          ***** 1.0 WITH SIX ASTERISKS.
C
      IF(MACH(IDX) - 0.3) 80,80,81
   81 IF(MACH(IDX) - 1.0) 82,83,83
   82 MFLG(IDX) = JAST1
      GO TO 80
   83 MFLG(IDX) = JAST6
   80 CONTINUE
      IF (DIAG(1,6HCOMFLO)) WRITE (IOT,6010) I1,GMA,CP,FLDMAX,M1,M2,DELP
      RETURN
C
C          ***** DIAGNOSTIC MESSAGE
C
 6000 FORMAT('0 *DIAGNOSTIC* MACH NUMBER COMPUTATION FAILED TO CONVERGE
     1WHILE COMPUTING COMPRESSABLE FLOW EQUATIONS, MACH SET = ',E12.7)
C
 6010 FORMAT ('+'14X,I5,10X,6E15.6)
      END
```

******** SUBRØUTINE CØMPIL

```
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C                    * ROUTINE NAME - DATA INPUT, VERIFY AND ECHO *
C                    * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                    * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                    * DATE CODED   - APRIL FOOLS DAY 1970         *
C                    * REVISED      - JANUARY 11,1972              *
C                    * REVISED      - JULY  1972                   *
C                    * PROGRAMMER   - J. MCKAY  D1943  201  45178 *
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE COMPIL
C
      LOGICAL JP,PAGE
C
      INCLUDE CACCUM
      INCLUDE CAPU
      INCLUDE CCNFIG
      INCLUDE CCNTRL
      INCLUDE CDCYCL
      INCLUDE CECLSS
      INCLUDE CENG
      INCLUDE CFUEL
      INCLUDE CHEX
      INCLUDE CHSORC
      INCLUDE CIOUNT
      INCLUDE CMOTOR
      INCLUDE CNAMES
      INCLUDE CPAGE
      INCLUDE CPUMP
      INCLUDE CTANK
      INCLUDE CTURBN
      INCLUDE  TANKWT
C
 5010 FORMAT(A6,I4,3I5,3F5.0,I5,2F5.0,5X,A6)
 5020 FORMAT(I5,6F10.0)
 5030 FORMAT(3I5/7F10.0/F10.0)
 5039 FORMAT(I5)
 5040 FORMAT(11F6.0,6X,A6)
 5050 FORMAT(I5,4F10.0/I5,4F10.0/4F10.0/4F10.0/5F10.0/5F10.0)
 5060 FORMAT (5I5/8F10.0/8F10.0/F10.0)
 5062 FORMAT (2I5,7F10.0)
 5070 FORMAT(F10.0)
 5080 FORMAT(3F10.0,I5,3F10.0,F7.0)
 5090 FORMAT(I5,5X,4F10.0)
 5100 FORMAT(I5,5X3F10.0)
 5110 FORMAT(8F10.0)
 5120 FORMAT(I5,5X,5F10.0)
 5130 FORMAT(7F10.0)
 5140 FORMAT(10F7.0)
 5141 FORMAT(2I5,6F10.0)
 5150 FORMAT(4I5,5F10.0/(7F10.0))
 5151 FORMAT(7F10.0/5F10.0)
C
 6000 FORMAT('0',38X9A6//21A6/21A6/' ')
 6010 FORMAT(3XA6, 2XA6,1XI4,2XI5,2XI5,2XI5,2X,E15.8,6XF7.2,6XF6.2,5XI5,
     1 9XF6.2,7XF5.1)
 6020 FORMAT('0 *DIAGNOSTIC* THE ABOVE FUNCTION CODE IS ILLEGAL')
```

```
********    COMPIL    *********

      6030 FORMAT('0',38X9A6/' ')
      6031 FORMAT('0',T55,'SUBCRITICAL APU DATA'//)
      6032 FORMAT('0',T53,'SUPERCRITICAL APU DATA'//)
      6040 FORMAT(54XI5,2X3A6/(46XE13.8,2X3A6))
      6050 FORMAT('0',38X9A6//)
      6051 FORMAT(T18,'- 1 -',T36,'- 2 -',T54,'- 3 -',T72,'- 4 -',T90,'- 5 -'
          1 ,T103,'HEAT EXCHANGER NUMBER'/T13,'OXYGEN  HYDROGEN  OXYGEN  HYDR
          2OGEN  OXYGEN  HYDROGEN  OXYGEN  HYDROGEN  OXYGEN  HYDROGEN'/)
      6052 FORMAT(/////T18,'- 6 -',T36,'- 7 -',T54,'- 8 -',T72,'- 9 -',T90,
          1 '- 10 -',T103,'HEAT EXCHANGER NUMBER'/T13,'OXYGEN  HYDROGEN  OXYG
          2EN  HYDROGEN  OXYGEN  HYDROGEN  OXYGEN  HYDROGEN  OXYGEN  HYDROGEN
          3'/)
      6060 FORMAT (41XI5,10XI5,7X4A6)
      6062 FORMAT ('0'//38X'*'9A6/ 39X'TANK'8X'FLUID'9X'. . . D I M E N S I O
          1 N S . . . .'/ 39X'SHAPE'7X'TYPE'22X'(FEET)'/)
      6064 FORMAT (30X 2I12,6X 3F12.4)
      6069 FORMAT(T48,'NUMBER OF HEAT EXCHANGERS INPUT =',I5//)
      6070 FORMAT(9X10F9.1,3X4A6)
      6080 FORMAT(42XI4,10XI5,7X4A6/4(36X2E15.8,2X4A6/),'0',38X9A6//4(36X2E1
          15.8 ,2X4A6/),'0',38X9A6/// (36X2E15.8,2X4A6))
      6090 FORMAT (36X2E15.8,2X4A6)
      6091 FORMAT('0',47X,I5,15X,'TANK WEIGHT-CONFIGURATION OPTION CONSIDERED
          1 '/48X,I5,15X,'NUMBER OF TANK SHAPES IN CONFIGURATION')
      6100 FORMAT('0'38X9A6/' ')
      6101 FORMAT('0',T10,'OPER.TIME',T24,'NON-OPERATING',T40,'MIB-DEGRAD.',
          1 T54,'UNITS OPER.',T67,'HORSEPOWER',T81,'AMB.PRESSURE',T98,
          2 'POWER-KW',T110,'REPRES.TIME'/)
      6110 FORMAT(T7,3E15.8,I8,5X,4E15.8)
      6120 FORMAT ('0 * ERROR * DUTY CYCLE INPUT TOO LONG'/)
      6127 FORMAT(//T18,'- 6 -',T36,'- 7 -',T54,'- 8 -',T72,'- 9 -',T90,
          1 '- 10 -',T103,'HEAT SOURCE NUMBER'/T13,
          2                                       'OXYGEN  HYDROGEN  OXYGEN  HYDR
          3OGEN  OXYGEN  HYDROGEN  OXYGEN  HYDROGEN  OXYGEN  HYDROGEN'/)
      6128 FORMAT(T49,'NUMBER OF HEAT SOURCES INPUT =',I5//)
      6129 FORMAT(T18,'- 1 -',T36,'- 2 -',T54,'- 3 -',T72,'- 4 -',T90,'- 5 -'
          1 ,T103,'HEAT SOURCE NUMBER'/T13,'OXYGEN  HYDROGEN OXYGEN  HYDROGEN
          2  OXYGEN  HYDROGEN  OXYGEN  HYDROGEN  OXYGEN  HYDROGEN'/)
      6130 FORMAT(9X10I9,3X4A6/(9X10F9.1,3X4A6))
      6140 FORMAT(T52,I15,T70,'MOTOR TYPE'/T52,E15.8,T70,'MOTOR EFFICIENCY',
          1 /T52,E15.8,T70,'MOTOR SPEED'/T52,E15.8,T70,'BATTERY POWER DENSITY
          2')
      6150 FORMAT(54XI5,2X4A6/(46XE13.8,2X4A6))
      6160 FORMAT(54XI5,2X4A6/54XI5,2X4A6/54XI5,2X4A6/54XI5,2X4A6,
          1 /(46XE13.8,2X4A6))
C
C  *****************************************************************
C
C       *****  CHECK 'NBLK' BEFOR DATA SETUP - INCLUDE ALL DATA BLOCKS  *****
C       *****  REQUIRED FOR THE SYSTEM BEING STUDIED.                   *****
C
C  *****************************************************************
C
        NIENTH = 0.0
C
C
C             ***** INPUT THE CONFIGURATION TABLE
C
        IF(PAGE(0)) WRITE(6,6000) (L0(I,1),I=1,9),((L1(I,J),I=1,21),J=1,2)
```

```
      JP = PAGE(5)
C
      IERR = 0
      DO 30 II=1,ICNF
      READ (IIN,5010) CFUNCT, CFTYPE, CNOPER, CNSTBY, CMTYPE, FRCOEF(II)
     1               , LOD(II), DIAM(II), CITYPE, ITHICK(II), NBAR(II),
     2                 CODE(II)
      IF(PAGE(1)) WRITE(6,6000) (LO(I,1),I=1,9),((L1(I,J),I=1,21),J=1,2)
      WRITE (IOT,6010) CFUNCT, CODE(II), CFTYPE, CNOPER, CNSTBY, CMTYPE,
     1                 FRCOEF(II), LOD(II), DIAM(II), CITYPE,
     2                 ITHICK(II), NBAR(II)
C
C     ***** SEARCH FOR THE CONFIGURATION NAME.
C
      DO 10 I2 = 1,18
      IF(CFUNCT.EQ.FNAME(I2)) GO TO 20
   10 CONTINUE
C
      WRITE(6,6020)
      IERR = IERR + 1
      GO TO 30
C
   20 CFUNCT = I2
      CALL STOCON(II)
      IF(CFUNCT.EQ.18) GO TO 35
   30 CONTINUE
C
C     ***** INPUT THE DUTY CYCLE DATA
C
   35 CONTINUE
      IF (IERR .GT. 0) CALL EXIT
      IF(PAGE(0)) WRITE(6,6100) (LO(I,6),I=1,9)
      WRITE (IOT,6101)
      DCYCLT = 0.0
      NDCYCL = 0
      II = 0
      DO 100 I1=1,ICDL2,2
      NDCYCL = NDCYCL + 2
      II = II + 1
      READ (IIN,5080) DCYCLE(I1), DCYCLE(I1+1), PSI(II), NEOP(II),
     1                HP(II), PAMB(II), PKW(II), RPRTIM(II)
      IF(PAGE(1)) WRITE(6,6100) (LO(I,6),I=1,9)
      WRITE (IOT,6110) DCYCLE(I1), DCYCLE(I1+1), PSI(II), NEOP(II),
     1                 HP(II), PAMB(II), PKW(II), RPRTIM(II)
      DCYCLT = DCYCLT + DCYCLE(I1)
      KCYCLE = II - 1
      IF(DCYCLE(I1+1)) 90,,
      IF(DCYCLE(I1)),100,100
      NDCYCL = NDCYCL - 1
   90 NDCYCL = NDCYCL - 1
      GO TO 110
  100 CONTINUE
C
      WRITE(6,6120)
      CALL EXIT
C
C     ***** INPUT THE CONSUMER DATA
C
```

```
  110 CONTINUE
      GO TO (111,112,120,130,111), SYSNUM
  111 CONTINUE
C
C          ***** READ IN THE ENGINE CONSUMER DATA
C
      IF(PAGE(0)) WRITE(6,6030) (LO(I,2),I=1,9)
      READ(5,5020) NENG,GITEMP,GIPRES,THRUST,PSUBC,EXPRAT,MIXRAT
      WRITE(6,6040) NENG,(L2(I,1),I=1,3),
     1            GITEMP,(L2(I,2),I=1,3),
     2            GIPRES,(L2(I,3),I=1,3),
     3            THRUST,(L2(I,4),I=1,3),
     4            PSUBC ,(L2(I,5),I=1,3),
     5            EXPRAT,(L2(I,6),I=1,3),
     6            MIXRAT,(L2(I,7),I=1,3)
C
      GO TO 113
C
  112 CONTINUE
C
C          ***  READ IN THE APU CONSUMER DATA
C
      IF(PAGE(0))WRITE (IOT,6030) (LO(I,13),I=1,9)
C
      READ (IIN,5120) NAPU, HPR, FMR, PGG, TIT, TD
      GO TO (114,115),SCRIT
  114 CONTINUE
      WRITE (IOT,6031)
      READ (IIN,5130) MRGGCH, MRGGCO, TDGGH, TDGGO, TVH, TVO, TENV
C
      WRITE (IOT,6150) NAPU, (L11(I,1),I=1,4), HPR, (L11(I,2),I=1,4),
     1                 FMR , (L11(I,3),I=1,4), PGG, (L11(I,4),I=1,4),
     2                 TIT , (L11(I,5),I=1,4), TD , (L11(I,6),I=1,4),
     3                MRGGCH,(L11(I,7),I=1,4),MRGGCO,(L11(I,8),I=1,4),
     4                TDGGH ,(L11(I,9),I=1,4),TDGGO ,(L11(I,10),I=1,4),
     5                TVH   ,(L11(I,11),I=1,4),TVO   ,(L11(I,12),I=1,4),
     6                TENV  ,(L11(I,13),I=1,4)
C
      GO TO 113
C
  115 CONTINUE
      WRITE (IOT,6032)
      READ (IIN,5130) FMRG, PFH, PFO, TFH, TFO, TG, DELPCP, TENV
C
      WRITE (IOT,6150) NAPU, (L11(I,1),I=1,4),HPR   , (L11(I,2),I=1,4),
     1                 FMR , (L11(I,3),I=1,4),PGG   , (L11(I,4),I=1,4),
     2                 TIT , (L11(I,5),I=1,4),TD    , (L11(I,6),I=1,4),
     3                 FMRG,(L11(I,14),I=1,4),PFH   ,(L11(I,15),I=1,4),
     4                 PFO ,(L11(I,16),I=1,4),TFH   ,(L11(I,17),I=1,4),
     5                 TFO ,(L11(I,18),I=1,4),TG    ,(L11(I,19),I=1,4),
     6               DELPCP,(L11(I,20),I=1,4),TENV  ,(L11(I,13),I=1,4)
C
      GO TO 113
C
  120 CONTINUE
C
C          *** READ IN LIFE SUPPORT SYSTEM CONSUMER DATA
C
```

```
      IF(PAGE(0)) WRITE (IOT,6050) (LO(I,15),I=1,9)
C
      READ (IIN,5150) MDAYS,NCREW,NRPRES,NDARES,O2FNOM,GLKRAT,TLSNOM(1),
     1 TLSNOM(2),RHOBEG(1),RHOBEG(2),TKFTEM(1),TKFTEM(2),TKFPRS(1),
     2 TKFPRS(2),TENVR,CABVOL
C
      READ (IIN,5151) LINDIA(1), LINDIA(2), HTRFLX(1), HTRFLX(2),
     1                PLSNOM(1), PLSNOM(2), HTRDIA(1), HTRDIA(2),
     2                HTRLNG(1), HTRLNG(2), PSET1, PSET2
C
      WRITE (IOT,6160) MDAYS,(L13(I, 1),I=1,4),
     1       NCREW    ,(L13(I, 2),I=1,4),NRPRES    ,(L13(I, 3),I=1,4),
     2       NDARES   ,(L13(I, 4),I=1,4),O2FNOM    ,(L13(I, 5),I=1,4),
     3       GLKRAT   ,(L13(I, 6),I=1,4),TLSNOM(1),(L13(I, 7),I=1,4),
     4       TLSNOM(2),(L13(I, 8),I=1,4),RHOBEG(1),(L13(I, 9),I=1,4),
     5       RHOBEG(2),(L13(I,10),I=1,4),TKFTEM(1),(L13(I,11),I=1,4),
     6       TKFTEM(2),(L13(I,12),I=1,4),TKFPRS(1),(L13(I,13),I=1,4),
     7       TKFPRS(2),(L13(I,14),I=1,4),TENVR    ,(L13(I,15),I=1,4),
     8       CABVOL   ,(L13(I,16),I=1,4),LINDIA(1),(L13(I,17),I=1,4),
     9       LINDIA(2),(L13(I,18),I=1,4),HTRFLX(1),(L13(I,19),I=1,4),
     T       HTRFLX(2),(L13(I,20),I=1,4),PLSNOM(1),(L13(I,21),I=1,4),
     A       PLSNOM(2),(L13(I,22),I=1,4),HTRDIA(1),(L13(I,23),I=1,4),
     B       HTRDIA(2),(L13(I,24),I=1,4),HTRLNG(1),(L13(I,25),I=1,4),
     C       HTRLNG(2),(L13(I,26),I=1,4),PSET1    ,(L13(I,27),I=1,4),
     D       PSET2    ,(L13(I,28),I=1,4)
C
      GO TO 113
C
  130 CONTINUE
C
C          *** READ IN FUEL CELL CONSUMER DATA
C
      IF(PAGE(0)) WRITE(IOT,6050) (LO(I,14),I=1,9)
C
      READ (IIN,5140) MRFC, SRCFC, QDTFC, SPWTFC, TFCNOM(1),TFCNOM(2),
     1                TF21IN, TF21OU, TFOFC, TFHFC
      READ (IIN,5140) PFOFC, PFHFC, RHOFIL(1), RHOFIL(2), WOVENT,WHVENT,
     1                DELTCP, TENV, PRFCOP, POWNOM
      READ (IIN,5141) NFCOP, NFCSTB, PLSET1, PLSET2, VJANUL(1),
     1                VJANUL(2),TKMXDI(1),TKMXDI(2)
      READ (IIN,5130) FCVOLT, PRGRAT(1), PRGRAT(2), PRGTIM(1),
     1                           PRGTIM(2), PRGINT(1), PRGINT(2)
C
      WRITE (IOT,6150) NFCOP   ,(L12(I, 1),I=1,4),MRFC,(L12(I, 2),I=1,4),
     1             SRCFC    ,(L12(I, 3),I=1,4),QDTFC     ,(L12(I, 4),I=1,4),
     2             SPWTFC   ,(L12(I, 5),I=1,4),TFCNOM(1),(L12(I, 6),I=1,4),
     3             TFCNOM(2),(L12(I, 7),I=1,4),TF21IN    ,(L12(I, 8),I=1,4),
     4             TF21OU   ,(L12(I, 9),I=1,4),TFOFC     ,(L11(I,18),I=1,4),
     5             TFHFC    ,(L11(I,17),I=1,4),PFOFC     ,(L11(I,16),I=1,4),
     6             PFHFC    ,(L11(I,15),I=1,4),RHOFIL(1),(L12(I,10),I=1,4),
     7             RHOFIL(2),(L12(I,11),I=1,4),WOVENT    ,(L12(I,12),I=1,4),
     8             WHVENT   ,(L12(I,13),I=1,4),DELTCP    ,(L11(I,20),I=1,4),
     9             TENV     ,(L11(I,13),I=1,4),PRFCOP    ,(L12(I,14),I=1,4),
     T             POWNOM   ,(L12(I,15),I=1,4),PLSET1    ,(L12(I,16),I=1,4),
     A             PLSET2   ,(L12(I,17),I=1,4),VJANUL(1),(L12(I,18),I=1,4),
     B             VJANUL(2),(L12(I,19),I=1,4),TKMXDI(1),(L12(I,20),I=1,4),
     C             TKMXDI(2),(L12(I,21),I=1,4),FCVOLT    ,(L12(I,22),I=1,4),
     D             PRGRAT(1),(L12(I,23),I=1,4),PRGRAT(2),(L12(I,24),I=1,4),
```

```
********   COMPIL   *********

     E          PRGTIM(1),(L12(I,25),I=1,4),PRGTIM(2),(L12(I,26),I=1,4),
     F          PRGINT(1),(L12(I,27),I=1,4),PRGINT(2),(L12(I,28),I=1,4)
C
C
  113 CONTINUE
C
C        ***** INPUT TANK DATA
C
      IF (PAGE(0)) WRITE (IOT,6050) (LO(I,12),I=1,9)
C
      J = 1
      READ (IIN,5060) (NOP    (I,J),SATYPE(I  ),SITYPE(I,J),SMTYPE(I,J),
     1      SPTYPE(I,J),SITEMP(I,J),SIPRES(I,J),SPGTEM(I,J),SOPRES(I,J),
     2      SVPRES(I,J),SHFLUX(I,J),SITHIK(I,J),FLDLOD(I  ),SULGPC(I  ),
     3      SMDIAM(I,J),SHOTEM(I,J),SHDELP(I,J),SPDELP(I,J),SGOTEM(I,J),
     4      SGGPC (I,J),SGMRAT(I,J),SNBAR (I  ),I=1,2)
      WRITE (IOT,6060) (NOP    (I,J),I=1,2),(L7(I,22),I=1,4),
     1                 (SATYPE(I  ),I=1,2),(L7(I,12),I=1,4),
     2                 (SITYPE(I,J),I=1,2),(L7(I,10),I=1,4),
     3                 (SMTYPE(I,J),I=1,2),(L7(I, 9),I=1,4),
     4                 (SPTYPE(I,J),I=1,2),(L7(I,13),I=1,4)
C
      WRITE (IOT,6090) (SITEMP(I,J),I=1,2),(L7(I, 6),I=1,4),
     1                 (SIPRES(I,J),I=1,2),(L7(I, 7),I=1,4),
     2                 (SPGTEM(I,J),I=1,2),(L7(I,15),I=1,4),
     3                 (SOPRES(I,J),I=1,2),(L7(I,14),I=1,4),
     4                 (SVPRES(I,J),I=1,2),(L7(I, 8),I=1,4),
     5                 (SHFLUX(I,J),I=1,2),(L7(I, 5),I=1,4),
     6                 (SITHIK(I,J),I=1,2),(L7(I,11),I=1,4),
     7                 (FLDLOD(I  ),I=1,2),(L7(I, 3),I=1,4),
     8                 (SULGPC(I  ),I=1,2),(L7(I, 4),I=1,4),
     9                 (SMDIAM(I,J),I=1,2),(L7(I, 2),I=1,4),
     A                 (SHOTEM(I,J),I=1,2),(L7(I,18),I=1,4),
     B                 (SHDELP(I,J),I=1,2),(L7(I,16),I=1,4),
     C                 (SPDELP(I,J),I=1,2),(L7(I,17),I=1,4),
     D                 (SGOTEM(I,J),I=1,2),(L7(I,20),I=1,4),
     E                 (SGGPC (I,J),I=1,2),(L7(I,19),I=1,4),
     F                 (SGMRAT(I,J),I=1,2),(L7(I,21),I=1,4),
     G                 (SNBAR (I  ),I=1,2),(L7(I,23),I=1,4)
C
      READ (IIN,5062) IWOP,NOSHAP
C
      WRITE (IOT,6091) IWOP, NOSHAP
C
C                 CHECK FOR GENERAL TANK CONFIGURATION
      IF (IWOP .LT. 2 .OR. NOSHAP .EQ. 0) GO TO 210
      WRITE (IOT,6062) (LO(I,10),I=1,9)
C                 READ IN GENERAL TANK CONFIGURATION
      DO 200 I=1,NOSHAP
      READ (IIN,5062) JTKTYP(I),JFLTP(I),XD(I),YD(I),ZD(I)
  200 WRITE(IOT,6064) JTKTYP(I),JFLTP(I),XD(I),YD(I),ZD(I)
  210 CONTINUE
C
      DO 2000 JSIM=1,5
C
      IF (INBLK(SYSNUM,JSIM,SCRIT) .EQ. 0) GO TO 2000
      GO TO (1100,1200,1300,1400,1500),JSIM
C
```

```
C          ***** INPUT THE ACCUMULATOR DATA
C
 1100 CONTINUE
      IF(PAGE(0)) WRITE(6,6050) (LO(I,3),I=1,9)
C
      READ (IIN,5030) (NAOP(I),AITYPE(I),AHTYPE(I),ATEMP(I),APRES(I),
     1      AHFLUX(I),AITHIK(I),AVOL(I),ADIAM(I),ANDELP(I),
     2      ANBAR (I),I=1,2)
C
      WRITE (IOT,6060) (NAOP  (I),I=1,2),(L7(I,22),I=1,4),
     1                 (AITYPE(I),I=1,2),(L7(I,10),I=1,4),
     2                 (AHTYPE(I),I=1,2),(L7(I, 9),I=1,4)
C
      WRITE (IOT,6090) (ATEMP (I),I=1,2),(L3(I, 1),I=1,4),
     1                 (APRES (I),I=1,2),(L7(I,14),I=1,4),
     2                 (AHFLUX(I),I=1,2),(L7(I, 5),I=1,4),
     3                 (AITHIK(I),I=1,2),(L7(I,11),I=1,4),
     4                 (AVOL  (I),I=1,2),(L3(I, 2),I=1,4),
     5                 (ADIAM (I),I=1,2),(L7(I, 2),I=1,4),
     6                 (ANDELP(I),I=1,2),(L3(I, 3),I=1,4),
     7                 (ANBAR (I),I=1,2),(L7(I,23),I=1,4)
      GO TO 2000
C
C          ***** INPUT THE HEX DATA
C
 1200 CONTINUE
      IF(PAGE(0)) WRITE(IOT,6050) (LO(I,4),I=1,9)
      READ(IIN,5039) NUMHEX
      READ(IIN,5040)((HEXHIT(I,J),HEXHOT(I,J),HEXCIT(I,J),HEXCOT(I,J),
     1                HEXHIP(I,J),HEXHOP(I,J),HEXCIP(I,J),HEXCOP(I,J),
     2                HXHDLP(I,J),HXCDLP(I,J),HXMRAT(I,J),HXCODE(I,J),
     3                J=1,2),I=1,NUMHEX)
      WRITE(IOT,6069) NUMHEX
      WRITE(IOT,6051)
      WRITE(IOT,6070)((HEXHIT(I,J),J=1,2),I=1,5),(L4(I, 1),I=1,4),
     1               ((HEXHOT(I,J),J=1,2),I=1,5),(L4(I, 2),I=1,4),
     2               ((HEXCIT(I,J),J=1,2),I=1,5),(L4(I, 3),I=1,4),
     3               ((HEXCOT(I,J),J=1,2),I=1,5),(L4(I, 4),I=1,4),
     4               ((HEXHIP(I,J),J=1,2),I=1,5),(L4(I, 5),I=1,4),
     5               ((HEXHOP(I,J),J=1,2),I=1,5),(L4(I, 6),I=1,4),
     6               ((HEXCIP(I,J),J=1,2),I=1,5),(L4(I, 7),I=1,4),
     7               ((HEXCOP(I,J),J=1,2),I=1,5),(L4(I, 8),I=1,4),
     8               ((HXHDLP(I,J),J=1,2),I=1,5),(L4(I, 9),I=1,4),
     9               ((HXCDLP(I,J),J=1,2),I=1,5),(L4(I,10),I=1,4),
     T               ((HXMRAT(I,J),J=1,2),I=1,5),(L4(I,11),I=1,4)
      IF (NUMHEX .LE. 5) GO TO 2000
      WRITE (IOT,6052)
      WRITE(IOT,6070)((HEXHIT(I,J),J=1,2),I=6,10),(L4(I, 1),I=1,4),
     1               ((HEXHOT(I,J),J=1,2),I=6,10),(L4(I, 2),I=1,4),
     2               ((HEXCIT(I,J),J=1,2),I=6,10),(L4(I, 3),I=1,4),
     3               ((HEXCOT(I,J),J=1,2),I=6,10),(L4(I, 4),I=1,4),
     4               ((HEXHIP(I,J),J=1,2),I=6,10),(L4(I, 5),I=1,4),
     5               ((HEXHOP(I,J),J=1,2),I=6,10),(L4(I, 6),I=1,4),
     6               ((HEXCIP(I,J),J=1,2),I=6,10),(L4(I, 7),I=1,4),
     7               ((HEXCOP(I,J),J=1,2),I=6,10),(L4(I, 8),I=1,4),
     8               ((HXHDLP(I,J),J=1,2),I=6,10),(L4(I, 9),I=1,4),
     9               ((HXCDLP(I,J),J=1,2),I=6,10),(L4(I,10),I=1,4),
     T               ((HXMRAT(I,J),J=1,2),I=6,10),(L4(I,11),I=1,4)
```

```
********    COMPIL    ********

          GO TO 2000
C
C               ***** INPUT THE PUMP DATA
C
  1300 CONTINUE
          IF(PAGE(0)) WRITE(6,6050) (LO(I,5),I=1,9)
          READ(5,5050)(PTYPE(I),PEFF(I),PNPSH(I),PSSPED(I),EPDELP(I),I=1,2),
     1               (TPEFF(I),TPNPSH(I),TPDELP(I),TPWDOT(I),I=1,2),
     2               (TEFF(I),TITEMP(I),TOTEMP(I),TMRATO(I),TGGPC(I),
     3 I=1,2)
          WRITE(6,6080)(PTYPE (I),I=1,2),(L5(I,1),I=1,4),
     1                (PEFF  (I),I=1,2),(L5(I,2),I=1,4),
     2                (PNPSH (I),I=1,2),(L5(I,3),I=1,4),
     3                (PSSPED(I),I=1,2),(L5(I,4),I=1,4),
     4                (EPDELP(I),I=1,2),(L5(I,5),I=1,4),
     5                (LO(I,11),I=1,9),
     6                (TPEFF (I),I=1,2),(L6(I,1),I=1,4),
     7                (TPNPSH(I),I=1,2),(L6(I,2),I=1,4),
     8                (TPDELP(I),I=1,2),(L6(I,3),I=1,4),
     9                (TPWDOT(I),I=1,2),(L6(I,4),I=1,4),
     T                (LO(I,8),I =1,9),
     1                (TEFF  (I),I=1,2),(L9(I,1),I=1,4),
     2                (TITEMP(I),I=1,2),(L9(I,2),I=1,4),
     3                (TOTEMP(I),I=1,2),(L9(I,3),I=1,4),
     4                (TMRATO(I),I=1,2),(L9(I,4),I=1,4),
     5                (TGGPC (I),I=1,2),(L9(I,5),I=1,4)
          GO TO 2000
C
C               ***** INPUT THE HEAT SOURCE DATA
C
  1400 CONTINUE
          IF (PAGE(0)) WRITE (IOT,6100) (LO(I,7),I=1,9)
          READ (IIN,5039) NUMHSO
          READ (IIN,5090) ((HSTYPE(I,J),HSMRAT(I,J),HSOTEM(I,J),HSAEE (I,J),
     1                     HSPRES(I,J),J=1,2),I=1,NUMHSO)
          WRITE (IOT,6128) NUMHSO
          WRITE (IOT,6129)
          WRITE (IOT,6130) ((HSTYPE(I,J),J=1,2),I=1, 5),(L10(I,1),I=1,4),
     1                     ((HSMRAT(I,J),J=1,2),I=1, 5),(L10(I,2),I=1,4),
     2                     ((HSOTEM(I,J),J=1,2),I=1, 5),(L10(I,3),I=1,4),
     3                     ((HSAEE (I,J),J=1,2),I=1, 5),(L10(I,4),I=1,4),
     4                     ((HSPRES(I,J),J=1,2),I=1, 5),(L10(I,5),I=1,4)
          IF (NUMHSO .LE. 5) GO TO 2000
          WRITE (IOT,6127)
          WRITE (IOT,6130) ((HSTYPE(I,J),J=1,2),I=6,10),(L10(I,1),I=1,4),
     1                     ((HSMRAT(I,J),J=1,2),I=6,10),(L10(I,2),I=1,4),
     2                     ((HSOTEM(I,J),J=1,2),I=6,10),(L10(I,3),I=1,4),
     3                     ((HSAEE (I,J),J=1,2),I=6,10),(L10(I,4),I=1,4),
     4                     ((HSPRES(I,J),J=1,2),I=6,10),(L10(I,5),I=1,4)
          GO TO 2000
C
C               ***** INPUT THE MOTOR DATA
C
  1500 CONTINUE
          IF (PAGE(0)) WRITE (IOT,6100) (LO(I,9),I=1,9)
          READ(5,5100) MTYPE,MEFF,MSS,PDNSTY
          WRITE (IOT,6140) MTYPE,MEFF,MSS,PDNSTY
C
```

```
*********    COMPIL    *********

      2000 CONTINUE
C
           RETURN
           END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - CØNST

```
CONST*  PROC
C
         COMMON /CONST/ GRAVTY,PI,PI2O3
C
C
C             ***** CONST VARIABLE DEFINITION
C                   *
C                   * GRAVTY = UNIVERSAL GRAVITATIONAL CONSTANT
C                   *
C
 END
```

******** SUBRØUTINE CØNSUM

```
C
      SUBROUTINE CONSUM
C
      INCLUDE CCNTRL
C
      GO TO (10,20), SYSNUM
C
   10 CALL ENGINE
C
      CALL FLORAT
C
      GO TO 99
C
   20 CALL APUFLO
C
      CALL FLORAT
C
      GO TO 99
C
   99 CONTINUE
C
      RETURN
      END
```

********* MAIN DRIVER PRØGRAM - CØNTROL

```
C
      LOGICAL PAGE
C
      INCLUDE CCNTRL
      INCLUDE CIOUNT
      INCLUDE CKEYS
      INCLUDE CPAGE
C
      DATA NSPC / 'SUP' / ILST / 'LAS' /
C
 5000 FORMAT(2A6,2XA4,3XA3,1XA5/12A6)
 5001 FORMAT(A3,A6,3X,A3,14X,A3,37X,'1')
 6000 FORMAT ('0'19X '***  SYSTEM NAME STARTING 'A3,' IS IN ERROR  ***')
 6001 FORMAT ('0'/// 44X'***  YOU HAVE CALLED FOR THE SYSTEM 'A3,A6,
     1 '***')
C
C                    INITIALIZE DATA STORAGE ROUTINES
C
      CALL STODTA
      CALL OTUNIT (IOT)
C
C
      CALL DATE (9,DOR)
C
C                    READ NAME AND TITLE INFORMATION
C
      READ(5,5000) NAME,DEPT,BLD,EXT,CTITLE
      NCASE = 1
      INTGSY = 1
C
C                    READ TABLE INPUT DATA  (ONLY ONCE)
C
      CALL INTAB
C
    1 CONTINUE
      KEY1 = 1
      KEY2 = 0
      IF(INTGSY.EQ.1) GO TO 5
      READ (5,5000) NAME,DEPT,BLD,EXT,CTITLE
    5 CONTINUE
      READ (IIN,5001) NSYS,NI,NCRIT,INTGR,MDTRC
      DO 10 I=1,NBRSY
      IF (NSYS .EQ. NAMSYS(I)) GO TO 20
   10 CONTINUE
      WRITE (IOT,6000) NSYS
      CALL EXIT
   20 SYSNUM = I
      IF (PAGE(0)) WRITE (IOT,6001) NSYS,NI
      SCRIT = 1
      IF (NCRIT .EQ. NSPC) SCRIT = 2
      INTGSY = 2
      IF (INTGR .EQ. ILST) INTGSY = 1
C
C                    READ COMPONENT INPUT DATA
C
      CALL COMPIL
C
```

```
********    CONTRL    ********

C                    DO SYSTEM CALCULATIONS FOR THE GIVEN SYSTEM
C
      CALL CRYCON
C
C                    IF THIS IS AN INTEGRATED SYSTEM - GO READ NEXT
C                    SYSTEM NAME
C                    IF IT IS A LAST CARD EXIT
C
      IF (INTGSY .EQ. 2) GO TO 1
      CALL EXIT
      END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR – CPAGE

```
CPAGE*  PROC
C
      INTEGER DEPT  ,BLD   ,EXT   ,PTITLE,CTITLE,OPTLUN,DOR   ,TIME
C
      COMMON /CPAGE/NAME(2),DEPT,BLD,EXT,MAXLIN,JNUM,PTITLE(6),
     1              CTITLE(12),OPTLUN,NCASE,DOR(2),TIME(2)
C
C
C        ***** CPAGE VARIABLE DEFINITION
C
C              NAME   - USER'S NAME (INPUT).
C
C              DEPT   - USER'S DEPARTMENT (INPUT).
C
C              BLD    - USER'S BUILDING (INPUT).
C
C              EXT    - USER'S EXTENSION (INPUT).
C
C              MAXLIN - MAXIMUM LINES PER OUTPUT PAGE.
C
C              JNUM   - PROGRAM IDENTIFICATION NUMBER.
C
C              PTITLE - PROGRAM TITLE.
C
C              CTITLE - CASE TITLE (INPUT).
C
C              OPTLUN - OUTPUT LOGICAL UNIT NUMBER.
C
C              NCASE  - CASE NUMBER (INTERNALLY CREATED).
C
C              DOR    - DATE OF THIS RUN (INTERNALLY CREATED).
C
C              TIME   - TIME OF THIS RUN (INTERNALLY CREATED).
C
 END
```

********* FUNCTIØN CPHI

```
      FUNCTION CPHI(T)
      COMMON /RFPR/ RF(10)
      COMMON /CIGCP/G(9)
      COMMON /SCRH/ X(40)
C
C.... ROUTINE TO CALCULATE INTEGRAL(CPO DT)
C
      R=RF(5)
      AK=RF(6)
      T2=T*T
      T3=T2*T
      T4=T3*T
      U=G(9)/T
      X(1) = -1.0/(2.0 * T2)
      X(2) = -1.0/T
      X(3)= ALOG(T)
      X(4)= T
      X(5)= T2/2.0
      X(6)= T3/3.0
      X(7)= T4/4.0
      X(8)= U*T/(EXP(U)-1.0)
      CPHI = 0.0
      DO 1 I=1,8
      CPHI=CPHI+X(I)*G(I)
      CPHI=CPHI*R
      CPHI=CPHI*AK
      RETURN
      END
```

******** FUNCTIØN CPIG

```
      FUNCTION CPIG(T)
C
      DOUBLE PRECISION G9, U, U2, DU
C
      COMMON /RFPR/ RF(10)
      COMMON /CIGCP/G(9)
      COMMON /SCRH/ X(40)
C
C.... CALCULATE IDEAL GAS HEAT CAPACITY, CP
C
      R=RF(5)
      AK=RF(6)
      T2=T*T
      T3=T2*T
      G9 = DBLE(G(9))
      U  = G9/DBLE(T)
      U2 = U * U
      X(1) = 1.0/T3
      X(2) = 1.0/T2
      X(3) = 1.0/T
      X(4) = 1.0
      X(5) = T
      X(6) = T2
      X(7) = T3
      DU = DEXP(U)
      X(8) = SNGL( U2 * DU/(DU-1.0D0)**2)
      CPIG = 0.0
      DO 1 I=1,8
    1 CPIG=CPIG+X(I)*G(I)
      CPIG=CPIG*R*AK
      RETURN
      END
```

******** FUNCTIØN CPSI

```
      FUNCTION CPSI(T)
      COMMON /CIGCP/G(9)
      COMMON /REFPR/ RF(10)
      COMMON /SCRH/ X(40)
C
C.... ROUTINE TO CALCULATE INTEGRAL(CPO/T DT)
C
      R=RF(5)
      AK=RF(6)
      T2=T*T
      T3=T2*T
      U=G(9)/T
      EU   = EXP(U)
      X(1) = -1.0/(3.0*T3)
      X(2) = -1.0/(2.0*T2)
      X(3) = -1.0/T
      X(4) = ALOG(T)
      X(5) = T
      X(6) = T2/2.0
      X(7) = T3/3.0
      X(8) = U/(EU-1.0) - ALOG(1.0-1.0/EU)
      CPSI = 0.0
      DO 1 I=1,8
      CPSI=CPSI+X(I)*G(I)
      CPSI=CPSI*R
      CPSI=CPSI*AK
      RETURN
      END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - CPUMP

```
CPUMP* PROC
C
      INTEGER PTYPE, PSTAGE
C
      COMMON /CPUMP/  EPDELP(2),PEFF  (2),PPDCH (2),PSSPED(2),PTYPE (2),
     1      TPDELP(2),TPEFF (2),TPNPSH(2),TPWDOT(2),TPWGHT(2),PTEMP (2),
     2      PPRES (2),PPWDT (2),PPDEL (2),PNPSH (2),PPRHO (2),PSTAGE(2),
     3      PNPSPR(2),PMPEFF(2),PMPVOL(2),PWEGHT(2),PMPOW (2),PSPD  (2),
     4      JJOPT (2),
     5      LPP1(3,6), LPP2(4), LPP3(3,6), LPP4(4), LPP5(4)
C
      DIMENSION  UOTP1(12),UOTP2(12)
C
      EQUIVALENCE          (UOTP1    ,PTEMP ),(UOTP1( 3),PPRES ),
     1   (UOTP1( 5),PPWDT ),(UOTP1( 7),PPDEL ),(UOTP1( 9),PNPSH ),
     2   (UOTP1(11),PPRHO ),(UOTP2    ,PNPSPR),(UOTP2( 3),PMPEFF),
     3   (UOTP2( 5),PMPVOL),(UOTP2( 7),PWEGHT),(UOTP2( 9),PMPOW ),
     4   (UOTP2(11),PSPD  )
C
C        ***** CPUMP VARIABLE DEFINITION.
C              *
C              * PEFF   - PUMP EFFICIENCY.
C              *
C              * PMPEFF - PUMP EFFICIENCY
C              *
C              * PMPVOL - PUMP VOLUME
C              *
C              * PMPOW  - PUMP POWER
C              *
C              * PWEGHT - PUMP WEIGHT  (LBS)
C              *
C              * PSPD   - PUMP SPEED
C              *
C              * PSTAGE - NUMBER OF PUMP STAGES
C              *
C              * PNPSPR - COMPUTED NPSH REQUIRED BY PUMP
C              *
C              * PTEMP  - TEMP OF FLUID TO PUMP
C              *
C              * PPRES  - PRESSURE OF FLUID FROM PUMP
C              *
C              * PPWDT  - FLOW RATE OF FLUID TO PUMP
C              *
C              * PPDEL  - PUMP DELTA-P
C              *
C              * PPRHO  - DENSITY OF FLUID TO PUMP
C              *
C              * JJOPT  - SELECTED OPTION FOR PUMP CAPABILITY, EQUALS
C              *          1 FOR MIN. POWER, EQUALS 2 FOR MIN. WEIGHT
C              *
C              * PTYPE  - PUMPTYPE  =  1 FOR PUMP ONLY
C              *                    =  2 FOR TURBOPUMP ASSY.
C              *
C              * TPEFF  - TRANSFER PUMP EFFECIENCY
C              *
C              * TPNPSH - TRANSFER PUMP NPSH
C              *
```

```
*********    CPUMP    *********

      C               * TPWGHT - TRANSFER PUMP WEIGHT
      C               *
      C               * TPDELP - TRANSFER PUMP DELTA-P
      C               *
      C               * TPWDOT - TRANSFER PUMP FLOW RATE
      C               *
      C               *
      C               * PNPSH  - PUMP NET POSITIVE SUCTION HEAD.
      C               *
      C               * PSSPED - PUMP SHAFT SPEED.
      C               *
      C               * EPDELP - ESTIMATED DELTA PRESSURE IN THE PUMP.
      C               *
      C               *  PPDCH  - PUMP DISCHARGE PRESSURE
      C               *
      C               * NOTE --- THE ABOVE VARIABLES ARE SPECIFIED FOR OXYGEN
      C               *          IN THE FIRST WORD AND HYDROGEN IN THE SECOND
      C           *****          WORD.
      C
       END
```

******** SUBRØUTINE CPVTD

```
      SUBROUTINE CPVTD(T,D,CP,CV)
      COMMON /CRPR/ CR(3) /METH/ M
      COMMON /RFPR/ RF(10)
C
C.... ROUTINE TO CALCULATE CV AND CP FROM THE EQUATION OF STATE
C
      DC=CR(2)
      TC=CR(3)
      R =RF(5)
      AK=RF(6)
      CALL PFND(T,D,P)
      IF(M.EQ.1)GO TO 1
      IF(T.GT.TC)GO TO 1
      IF(D.GT.DC)GO TO 2
    1 CVO=CPIG(T)-(R*AK)
      CV = CVO - (FING3(T,D)-FING3(T,0.0)) * AK
    3 F1=T/D**2
      F2=DPDT(T,D)
      F3=DPDD(T,D)
      CP=CV+(F1*F2**2/F3)*AK
      RETURN
    2 DT = 0.1
      T1=T+DT
      T2=T-DT
      CALL LPROP(T1,P1,D,1,H,S,U1,Z)
      CALL LPROP(T2,P2,D,1,H,S,U2,Z)
      CV = (U1-U2)/(2.0*DT)
      GO TO 3
      END
```

******** SUBRØUTINE CPVTDB

```
      SUBROUTINE CPVTDB(TB,DB,CPB,CVB)
      COMMON /RFPR/ RF(8)
      WT=RF(7)
      T  = TB/1.8
      D  = DB * 453.59237E-3/(WT * 2.8316847E-2)
      CALL CPVTD(T,D,CP,CV)
      CPB = CP * 453.59237/(1.0543503E+3 * 1.8 * WT)
      CVB = CV * 453.59237/(1.0543503E+3 * 1.8 * WT)
      RETURN
      END
```

******** SUBRØUTINE CRYCØN

```
C
      SUBROUTINE CRYCON
C
      LOGICAL DIAG,JP
C
      INCLUDE CCNTRL
      INCLUDE CKEYS
C
      KEY1 = 1613
C
      K = 0
C
      IF(SYSNUM.NE.2) GO TO 10
      DO 6 I = 1,2
    6 KSUBC(2,I+1) = JAPUS(SCRIT,I)
C
   10 I = 1
      K = 0
      LREPT = 0
C
   20 JKM = KSUBC(SYSNUM,I)
C
C                    IF ZERO HAVE REACHED END OF CALLING SEQUENCE
      IF (JKM .EQ. 0) GO TO 2200
C                    MDTRC - IS DIAGNOSTIC TRACE SWITCH (INPUT IN CONTRL)
      IF (MDTRC(JKM) .EQ. 0) GO TO 50
C                    TURN ON DIAG. TRACE
      JP = DIAG (-1,6HCRYCON)
C                    CALL ROUTINES DEPENDING ON SYSTEM SPECIFIED (SYSNUM)
   50 GO TO (100,200,300,400,450,500,550,600,700,800,900,1000),JKM
C
  100 CALL ACCRES
      GO TO 2000
C
  200 CALL ACQWT
      GO TO 2000
C
  300 GO TO (310,350),SCRIT
C
  310 CALL APUSUB
      GO TO 2000
C
  350 CALL APUSUP
      GO TO 2000
C
  400 CALL CMPCAL
      GO TO 2000
C
  450 CALL FUELCL
      GO TO 2000
C
  500 CALL CONSUM
      GO TO 2000
C
  550 CALL ECLSS
      GO TO 2000
C
```

```
  600 CALL LIQRES
      GO TO 2000
C
  700 CALL TANK
      GO TO 2000
C
  800 K = K + 1
      IF (SYSNUM .EQ. 2) K = 2
C
      CALL TSIZEI (K)
      GO TO 2000
C
  900 CALL WTACC
C
 1000 CONTINUE
C
 2000 CONTINUE
      IF(LREPT) 10,2001,10
C
 2001 IF(MDTRC(JKII) .EQ. 0) GO TO 2100
C
C                         TURN OFF DIAGNOSTIC TRACE SENTINEL
C
      JP = DIAG (-2,6HCRYCON)
C
 2100 I = I + 1
      IF(I-9) 20,20,2200
C                         END OF PROCESS THIS SYSTEM
 2200 CONTINUE
C
C                         PRINT COMPONENT WEIGHT SUMMARY
      CALL OTPWSM
C
      RETURN
      END
```

********* FUNCTIØN CSPF21

```
C
      FUNCTION CSPF21 (TMP)
C
      DIMENSION T(9),CP(9)
C
      DATA T / 415.0,465.0,510.0,545.0,575.0,590.0,610.0,660.0,700.0 /
      DATA CP/ .2335,.2380,.2455,.2530,.2630,.2700,.2800,.3160,.3520 /
C
      DO 10 I=1,9
      J = I
      IF (T(J) .GE. TMP) GO TO 20
   10 CONTINUE
C
   20 IF (J .EQ. 1) J = 2
      I = J - 1
      CSPF21 = CP(I) + (TMP - T(I))*(CP(J) - CP(I))/(T(J) - T(I))
      RETURN
      END
```

******** SUBRØUTINE CSUBP

```
      SUBROUTINE CSUBP(TEMP,PRES,NGAS,C)
C
C     *******************************************************************
C
C     NGAS  =  1    FOR OXYGEN
C           =  2    FOR HYDROGEN
C           =  17   FOR HELIUM
C           =  18   FOR NITROGEN
C
C     *******************************************************************
C
      COMMON/CVALUE/ CPON, CVON, DON
C
      T = TEMP
      P = PRES
      TTT = TEMP
      PPP = PRES
C
      GO TO (1,2,5,5,5,5,5,5,5,5,5,5,5,5,5,5,17,18),NGAS
C
C     COMPUTE CSUBP FOR OXYGEN
C
    1 CALL CSUBPV(TTT,PPP,1)
      C = CPON
      RETURN
C
C     COMPUTE CSUBP FOR HYDROGEN
C
    2 C = HPTCP(P,T)
      RETURN
C
C     FOR ERRONEOUS NGAS INDEX
C
    5 C = 0.0
      WRITE (6,100)
  100 FORMAT('0',20('*'),'  ERRONEOUS NGAS INDEX WAS USED IN S.R. CSUBP
     1 ',20('*')/)
      RETURN
C
C     COMPUTE CSUBP FOR HELIUM
C
   17 TR = T/17.37
      PR = P/41.82
       IF(TR.GT.1.6460845) GO TO 22
       POWER     =1.-4.62*ALOG10(TR)
       GO TO 24
   22  POWER=.5930735918-2.74*ALOG10(TR)
   24  ACONST=10.**POWER
      C = 1.252 + ACONST * PR * 386.3/1544.0
C
      RETURN
C
C     COMPUTE CSUBP FOR NITROGEN
C
   18 CALL CSUBPV(TTT,PPP,18)
      C = CPON
      RETURN
```

******** CSUBP ********

```
C
      END
```

******** SUBRØUTINE CSUBPV

```
      SUBROUTINE CSUBPV(T,P,NGAS)
C
      COMMON /CRPR/ CR(3) /METH/ M
      COMMON /RFPR/ RF(8)
      COMMON/CVALUE/ CPON, CVON, DON
C
      IF(NGAS.EQ.1) KF = 1
      IF(NGAS.EQ.18) KF = 2
      IF(KF.EQ.1)CALL DATAO2
      IF(KF.EQ.2)CALL DATAN2
C
C     KF =1 INDICATES OXYGEN
C     KF =2 INDICATES NITROGEN
C
      M = 1
C
      CALL DFNDB(T,P,D,Z1,0)
      DON = D
C
      CALL CPVTDB(T,D,CP,CV)
      CPON = CP
      CVON = CV
C
C
      RETURN
C
      END
```

********* SUBRØUTINE CSUBPI

```
      SUBROUTINE CSUBPI(T,R,C)
C
C  COMPUTE SPECIFIC HEAT CAPACITY (IN BTU/LB-DEG R) AT CONSTANT PRESSURE
C  FROM THE GRAPHICAL DATA OF THE COMBUSTION PRODUCTS OF H2 AND O2 AS
C  A FUNCTION OF MIXTURE RATIO AND TURBINE INLET TEMPERATURE.
C
C  T = TURBINE INLET TEMPERATURE
C  R = WEIGHT RATIO OF O2 TO H2
C  C = COMPUTED VALUE OF SPECIFIC HEAT OF EXHAUST GAS AT CONSTANT PRES.
C
      DIMENSION X(9), CP(10,9),  TT(10)
C
C  VALUES OF MIXTURE RATIO FOR WHICH CP IS TABULATED.
C
      DATA (X(I),I=1,9)/.5, .75, 1.0, 1.5, 2.0, 3.0, 4.0, 6.0, 8.0/
C
C  VALUES OF TEMPERATURE FOR WHICH CP IS TABULATED.
C
      DATA (TT(I), I=1,10)/ 1400.0, 1500.0, 1600.0, 1700.0, 1800.0,
     1                      1900.0, 2000.0, 2100.0, 2200.0, 2300.0/
C
C  TABULATED VALUES OF CP AS A FUNCTION OF MIXTURE RATIO AND TEMPERATURE
C
      DATA (CP(I,1),I=1,10)/ 2.40, 2.41, 2.425, 2.44, 2.46, 2.475, 2.495
     1, 2.515, 2.53, 2.545/
      DATA (CP(I,2),I=1,10)/  2.07, 2.08, 2.1, 2.115, 2.13, 2.14, 2.155,
     1  2.175, 2.19, 2.21/
      DATA (CP(I,3),I=1,10)/ 1.83, 1.84, 1.855, 1.865, 1.88, 1.895, 1.91
     1, 1.92, 1.935, 1.955/
      DATA (CP(I,4),I=1,10)/ 1.485, 1.5, 1.51, 1.52, 1.535, 1.55, 1.56,
     1  1.575, 1.59, 1.605/
      DATA (CP(I,5),I=1,10)/ 1.25, 1.26, 1.275, 1.285, 1.295, 1.31,
     1  1.325, 1.335, 1.345, 1.36/
      DATA (CP(I,6),I=1,10)/ .965, .975, .985, .995, 1.005, 1.015, 1.025
     1, 1.035, 1.05, 1.06/
      DATA (CP(I,7),I=1,10)/ .8, .81, .82, .83, .84, .85, .86, .87, .88,
     1 .895/
      DATA (CP(I,8),I=1,10)/ .605, .615, .625, .635, .645, .655, .665,
     1 .675, .685, .695/
      DATA (CP(I,9),I=1,10)/ .5, .51, .52, .53, .54, .55, .56, .57, .58,
     1 .59/
C
C  DETERMINE THE INTERVAL CONTAINING TURBINE INLET TEMPERATURE.
C
      DO 100 II=2,10
      I = II - 1
      IF(T .LT. TT(II)) GO TO 150
  100 CONTINUE
C
C  DETERMINE THE INTERVAL CONTAINING THE MIXTURE RATIO.
C
  150 DO 200 II=2,9
      J = II - 1
      IF(R .LT. X(II)) GO TO 220
  200 CONTINUE
C
C  INTERPOLATE FIRST ON TEMPERATURE AND THEN ON MIXTURE RATIO.
```

```
*********     CSUBP1     *********

      C
        220 C1 =((CP(I+1,J) - CP(I,J))/(TT(I+1) - TT(I)))*(T - TT(I))
            C1 = C1 + CP(I,J)
            C2 = ((CP(I+1,J+1) - CP(I,J+1))/(TT(I+1) - TT(I)))*(T - TT(I))
            C2 = C2 + CP(I,J+1)
            C = ((C2 - C1)/(X(J+1) - X(J)))*(R - X(J)) + C1
            RETURN
            END
```

********* FUNCTIØN CSUBV

```
      FUNCTION CSUBV(TEMP,PRES,NGAS)
C
C     ******************************************************************
C
C     NGAS  =  1   FOR OXYGEN
C           =  2   FOR HYDROGEN
C           =  17  FOR HELIUM
C           =  18  FOR NITROGEN
C
C     ******************************************************************
C
      COMMON/CVALUE/ CPON, CVON, DON
C
      T = TEMP
      P = PRES
      TTT = TEMP
      PPP = PRES
C
      GO TO (1,2,5,5,5,5,5,5,5,5,5,5,5,5,5,5,17,18),NGAS
C
C     COMPUTE CSUBV FOR OXYGEN
C
    1 CALL CSUBPV(TTT,PPP,1)
      CSUBV = CVON
      RETURN
C
C     COMPUTE CSUBV FOR HYDROGEN
C
    2 CSUBV = HPTCV(P,T)
      RETURN
C
C     FOR ERRONEOUS NGAS INDEX
C
    5 CSUBV = 0.0
      WRITE (6,100)
  100 FORMAT('0',20('*'),'  ERRONEOUS NGAS INDEX WAS USED IN S.R. CSUBV
     1 ',20('*')/)
      RETURN
C
C     COMPUTE CSUBV FOR HELIUM
C
   17 CSUBV = 0.746
      RETURN
C
C     COMPUTE CSUBV FOR NITROGEN
C
   18 CALL CSUBPV(TTT,PPP,18)
      CSUBV = CVON
C
      RETURN
C
      END
```

********   PRØCEDURE DEFINITIØN PRØCESSØR - CSYSWT

```
CSYSWT* PROC
C
      PARAMETER  JWA=100, JWB=5
C
      INTEGER  CNUM
C
C
      COMMON /CSYSWT/ CNUM, WTCOMP(JWA,JWB), WTCUMT(JWA,JWB),
     1     CMPCOD(JWA,JWB), WTSUBT         , WTSUMT
C
C          ***** CNUM   -  COMPONENT INDEX IN SERIAL LIST
C              *
C              * WTCOMP -  COMPONENT WEIGHT
C              *
C              * WTCUMT -  SYSTEM CUMULATIVE WEIGHT IN SERIAL LIST
C              *
C              * CMPCOD -  COMPONENT ID CODE FROM CONFIGURATION
C              *
C              * WTSUBT -  SUBSYSTEM WEIGHT SUBTOTAL
C              *
C              * WTSUMT -  SYSTEM TOTAL WEIGHT
C          *****
C
 END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR - CTAB

```
CTAB*   PROC
C
      PARAMETER  NTBN = 50 , NSBZ = 40
C
      INTEGER TLA,TYPE
C
      COMMON /CTAB/ TLA(NTBN),NV,TYPE,NIP,ND,XTAB(NSBZ),YTAB(NSBZ),
     1 TAB(6,5),JTABID,NLTBL
C
      DIMENSION ITAB(6,5)
C
      EQUIVALENCE (ITAB,TAB)
C
C         ***** CTAB VARIABLE DEFINITION
C
C              TLA  - TABLE LOCATION ARRAY
C                     THIS ARRAY CONTAINS THE BEGINNING DRUM ADDRESS
C                     - 1, FOR UP TO 50 TABLES
C
C              NV   - NUMBER OF VALUES IN THE INPUT TABLE (NV<101).
C
C              TYPE - TYPE OF THE INPUT TABLE (0 = COEFICIENT,
C                                              1 = DESCRETE  ).
C
C              NIP  - NUMBER OF TABLE VALUES TO BE USED IN
C                     INTERPOLATION (NIP<NV).
C
C              ND   - NUMBER OF DIMENSIONS FOR THE INPUT TABLE
C                     (ND<7).
C
C              XTAB - ARRAY OF COEFICIENTS FOR POLYNOMIAL EVALUATION
C                     OR ARRAY OF VALUES OF THE INDEPENDENT VARIABLE
C                     FOR INTERPOLATION.
C
C              YTAB - ARRAY OF VALUES OF THE DEPENDENT VARIABLE
C                     FOR INTERPOLATION.
C
C               TAB - ARRAY OF VALUES OF THE REMAINING ND-2
C              ITAB   INDEPENDENT VARIABLES FOR INTERPOLATION.
       END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR – CTABA

```
CTABA*  PROC
C
      PARAMETER MXWRD = 7000
C
      COMMON /CTABA/ IDXI,TABLE(MXWRD)
C
      DIMENSION  ITABLE(MXWRD)
C
      EQUIVALENCE (TABLE,ITABLE)
C
 END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR – CTANK

```
CTANK*  PROC
C
      PARAMETER IJA=2,IJB=1,IJC=30
C
      INTEGER  SATYPE,SITYPE,SMTYPE,SPTYPE
C
      COMMON /CTANK/   FLDLOD(IJA    ),NOP    (IJA,IJB),SATYPE(IJA    ),
     1 SGGPC (IJA,IJB),SGMRAT(IJA,IJB),SGOTEM(IJA,IJB),SHDELP(IJA,IJB),
     2 SHFLUX(IJA,IJB),SHOTEM(IJA,IJB),SHRATE(IJA,IJB),SIPRES(IJA,IJB),
     3 SITEMP(IJA,IJB),SITYPE(IJA,IJB),SITHIK(IJA,IJB),SMDIAM(IJA,IJB),
     4 SMTYPE(IJA,IJB),SOPRES(IJA,IJB),SPDELP(IJA,IJB),SPGTEM(IJA,IJB),
     5 SPTYPE(IJA,IJB),SULGPC(IJA    ),SVLFLD(IJA    ),SVOL  (IJA,IJB),
     6 SVPRES(IJA,IJB),SWVTOT(IJA,IJB),TIWT  (IJA,IJB),TSA   (IJA,IJB),
     7 TWT   (IJA,IJB),WBPG  (IJA,IJB),WCPPG (IJA,IJB),WGGAPG(IJA,IJB),
     8               SNBAR (IJA    ),WGGP  (IJA,IJB),WGGPPG(IJA,IJB),
     9 WGGTOT(IJA    ),WGR   (IJA,IJB),WHETOT(IJA,IJB),WLR   (IJA    ),
     A WLRT  (IJA,IJB),WMPG  (IJA,IJB),                WPGTOT(IJA,IJB),
     B WPTOT (IJA    ),WTACQ (IJA,IJB),WTGGH2(IJA,IJB),WTGGO2(IJA,IJB),
     C WTHXPG(IJA    ),WTOTP (IJA    ),WTSYPG(IJA,IJB),WTTOT (IJA    ),
     D                     PRESHE(IJC,IJA,IJB),SVWT  (IJC,IJA,IJB),
     E WDOTHE(IJC,IJA,IJB),WDOTPG(IJC,IJA,IJB)
     F,LTZ1(3,14),INDXTK(IJA),LTZ2(2,3),LTZ3(3,4),TCYHT(IJA)
C
C         ***** CTANK VARIABLE DEFINITION
C              *
C              * SATYPE - ACQUISITION TYPE
C              *
C              * SITYPE - INSULATION TYPE
C              *
C              * SMTYPE - MATERIAL TYPE
C              *
C              * SPTYPE - PRESURIZATION TYPE
C              *
C              * SMDIAM - MAXIMUM DIAMETER (FT)
C              *
C              * SHFLUX - HEAT FLUX (BTU/HR-FT**2)
C              *
C              * SITEMP - INITIAL TEMP (DEG. R)
C              *
C              * SIPRES - INITIAL PRES (PSIA)
C              *
C              * SVPRES - VENTING PRES (PSIA)
C              *
C              * SITHIK - INSULATION THICKNESS (IN)
C              *
C              * SOPRES - OPERATING PRES (PSIA)
C              *
C              * SPGTEM - PRESSURANT GAS TEMP (DEG. R)
C              *
C              * SHDELP - HEX DELTA PRES (PSIA)
C              * SPDELP - PUMP DELTA PRES (PSIA)
C              *
C              * SHOTEM - HEX OUTLET TEMP (DEG. R)
C              *
C              * SGGPC  - P SUB C OF THE GAS GEN. (PSIA)
C              *
C              * SGOTEM - GAS GEN. OUTLET TEMP.(DEG. R)
```

```
********    CTANK    ********

      C             *
      C             * SNBAR  - NUMBER OF LAYERS OF INSULATION ON TANK
      C             *
      C             * WTOTP - TOTAL PROPELLANT IN TANK INITIALLY
      C             *
      C             * WLRT   - LIQUID RESIDUALS COMPUTED IN TANK
      C             *
      C         ***** SGMRAT - GAS GEN. MIXTURE RATIO
      C
        END
```

```
CTURBN* PROC
C
      COMMON /CTURBN/ TEFF (2),TGGPC (2),TITEMP(2),TMRATO(2),TOTEMP(2),
     1      TTRMD (2),TWPTEL(2),TWGTTR(2),TWMFNZ(2),TWNDCR(2),TWEGHT(2),
     2      WDGGFR(2),GWEGHT(2),WGTGGA(2),
     3      LTBN1(4),LTBN2(5,6)
C
      DIMENSION UOTB1(12),UOTB2(6)
C
      EQUIVALENCE           (UOTB1    ,TTRMD ),(UOTB1( 3),TWPTEL),
     1   (UOTB1( 5),TWGTTR),(UOTB1( 7),TWMFNZ),(UOTB1( 9),TWNDCR),
     2   (UOTB1(11),TWEGHT),(UOTB2    ,WDGGFR),(UOTB2( 3),GWEGHT),
     3   (UOTB2( 5),WGTGGA)
C
C          ***** CTURBN VARIABLE DEFINITION
C              *
C              * TMRATO - TURBINE MIXTURE RATIO.
C              *
C              * TEFF   - TURBINE EFFICIENCY.
C              *
C              * TITEMP - TURBINE INLET TEMPRATURE.
C              *
C              * TOTEMP - TURBINE OUTLET TEMPRATURE.
C              *
C              * TWEGHT - TURBINE WEIGHT
C              *
C              * GWEGHT - INTEGRATED GAS GEN. PROPELLANT WEIGHT
C              *
C              * WDGGFR - GAS GENERATOR FLOW RATE
C              *
C              * WGTGGA - GAS GENERATOR WEIGHT
C              *
C              * TGGPC  - TURBINE GAS GENERATOR CHAMBER PRESSURE
C              *
C              * TTRMD  - TURBINE ROTOR MEAN DIAMETER
C              *
C              * TWPTEL - WGT. POWER TRANSMISSION ELEMENT
C              *
C              * TWGTTR - WGT. TURBINE ROTOR
C              *
C              * TWMFNZ - WGT. INLET MANIFOLD AND NOZZLE
C              *
C              * TWNDCR - WGT. OF INDUCER ASSY.
C              *
C              * NOTE   - VARIABLES ARE SPECIFIED FOR OXYGEN
C              *          IN THE FIRST WORD AND HYDROGEN IN
C          *****          THE SECOND WORD
C
       END
```

******** SUBRØUTINE DATAN2

```
      SUBROUTINE DATAN2
      COMMON /CEOS/G(41) /CVPN/GV(11) /CIGCP/GI(9) /CSL/CL(7) /CSV/CV(7)
     .       /CRPR/CR(3) /CTEVP/CT(8) /RFPR/RF(10) /METH/ M
C
C.... IF THE PROPERTIES OF NITROGEN ARE TO BE CALCULATED A CALL TO THIS
C.... SUBROUTINE MUST BE THE FIRST CALL STATEMENT IN THE MAIN PROGRAM
C
C.... THE COMMON BLOCKS INITIALIZED IN THIS ROUTINE HOLD THE FOLLOWING
C.... INFORMATION -
C
C         /CEOS/ G(41)   COEFFICIENTS OF THE EQUATION OF STATE
C
C         /CIGCP/ CIG(9) COEFFICIENTS OF THE IDEAL GAS HEAT CAPACITY EQUATION
C
C         /CVPN/ GV(11)  COEFFICIENTS OF THE VAPOR PRESSURE EQUATION
C
C         /CRPR/ CR(3)   THE CRITICAL PROPERTIES IN THE SAME UNITS AS THE
C                        EQUATION OF STATE
C                        CR(1)=CRITICAL PRESSURE
C                        CR(2)=CRITICAL DENSITY
C                        CR(3)=CRITICAL TEMPERATURE
C
C         /CSL/ SL(7)    COEFFICIENTS OF EQUATION USED TO APPROXIAMATE
C                        THE SATURATED LIQUID DENSITY AS A FUNCTION OF
C                        TEMPERATURE
C
C
C         /RFPR/ RF(10)  REFERENCE PROPERTIES
C                        RF(1)=REFFERENCE ENTHALPY AT TEMPERATURE TOH
C                        RF(2)=REFFERENCE ENTROPY AT TEMPERATURE TOS
C                        RF(3)=TEMPERATURE TOH
C                        RF(4)=TEMPERATURE TOS
C                        RF(5)=GAS CONSTANT IN UNITS OF EQUATION OF STATE - R
C                        RF(6)=CONVERSION FACTOR TO CHANGE UNITS OF
C                              THE EQUATION OF STATE TO DESIRED ENERGY UNITS
C                        RF(7)=MOLECULAR WEIGHT
C                        RF(8)=TRIPLE POINT TEMPERATURE
C                        RF(9)  - NOT USED
C                        RF(10) - NOT USED
C
C         /METH/ M       INDICATES METHOD TO BE USED IN THE CALCULATION
C                        OF LIQUID PROPERTIES
C                        M=1  INDICATES ISOTHERM INTEGRATION THROUGH THE DOME
C                        M=2  INDICATES THE USE OF THE CLAPEYRON EQUATION
C                             THROUGH THE DOME
      M = 1
      G( 1) =  0.13622477E-02
      G( 2) =  0.10703247E+00
      G( 3) = -0.24390072E+01
      G( 4) =  0.34100745E+02
      G( 5) = -0.42237431E+04
      G( 6) =  0.10509860E-03
      G( 7) = -0.11259483E-01
      G( 8) =  0.14260079E-03
      G( 9) =  0.18469850E+05
      G(10) =  0.81114008E-07
      G(11) =  0.23301165E-02
```

B-100

LMSC-A991396

```
*********    DATAN2    *********

  58           G(12) =  -0.50775258E+00
  59           G(13) =   0.48502788E-04
  60           G(14) =  -0.11365676E-02
  61           G(15) =  -0.70743027E+00
  62           G(16) =   0.75170665E-04
  63           G(17) =  -0.11161412E-05
  64           G(18) =   0.36879656E-03
  65           G(19) =  -0.20131769E-05
  66           G(20) =  -0.16971744E+05
  67           G(21) =  -0.11971924E+06
  68           G(22) =  -0.97521827E+02
  69           G(23) =   0.55463971E+05
  70           G(24) =  -0.17992045E+00
  71           G(25) =  -0.25658293E+01
  72           G(26) =  -0.41370772E-03
  73           G(27) =  -0.25624542E+00
  74           G(28) =  -0.12422237E-06
  75           G(29) =   0.10355654E-04
  76           G(30) =  -0.53359917E-09
  77           G(31) =  -0.75741541E-08
  78           G(32) =   0.58536717E-07
  79           G(41) =  -0.56000000E-02
  80          GV( 1) =   0.83944094E+04
  81          GV( 2) =  -0.18900453E+04
  82          GV( 3) =  -0.72822292E+01
  83          GV( 4) =   0.00000000E+00
  84          GV( 5) =   0.55560638E-03
  85          GV( 6) =  -0.59445447E-05
  86          GV( 7) =   0.27154339E-07
  87          GV( 8) =  -0.48795359E-10
  88          GV( 9) =   0.50953608E+03
  89          GV(10) =   0.10228510E-01
  90          GV(11) =   0.19500000E+01
  91          GI( 1) =  -0.73521040E+03
  92          GI( 2) =   0.34223998E+02
  93          GI( 3) =  -0.55764828E+00
  94          GI( 4) =   0.35040423E+01
  95          GI( 5) =  -0.17339019E-04
  96          GI( 6) =   0.17465085E-07
  97          GI( 7) =  -0.35689203E-11
  98          GI( 8) =   0.10053872E+01
  99          GI(9) =    0.33534061E+04
 100          CL( 1) =   0.19424403E+02
 101          CL( 2) =   0.57083748E+02
 102          CL( 3) =  -0.24326985E+03
 103          CL( 4) =   0.88516838E+03
 104          CL( 5) =  -0.16393680E+04
 105          CL( 6) =   0.11574320E+04
 106          CL( 7) =   0.10182210E+01
 107          CV( 1) =   0.16333345E+01
 108          CV( 2) =  -0.94043771E+01
 109          CV( 3) =   0.21852746E+02
 110          CV( 4) =  -0.10268744E+03
 111          CV( 5) =   0.18794974E+03
 112          CV( 6) =  -0.16402437E+03
 113          CV( 7) =  -0.95731639E-01
 114          CR( 1) =   0.33555000E+02
 115          CR( 2) =   0.11210000E+02
```

```
********    DATAN2    ********

      CR( 3) =   0.12620000E+03
      CT( 1) =  -0.14206479E-02
      CT( 2) =   0.12908809E-01
      CT( 3) =   0.00
      CT( 4) =   0.00
      CT( 5) =   0.00
      CT( 6) =   0.00
      CT( 7) =   0.00
      CT( 8) =   0.00
      RF( 1) =   0.86690000E+04
      RF( 2) =   0.19150200E+03
      RF( 3) =   0.29815000E+03
      RF( 4) =   0.29815000E+03
      RF( 5) =   0.82053900E-01
      RF( 6) =   0.10132780E+03
      RF( 7) =   0.28013400E+02
      RF( 8) =   0.63148000E+02
      RETURN
      END
```

SUBRØUTINE DATAØ2

```
      SUBROUTINE DATA02
      COMMON /CEOS/G(41) /CVPN/GV(11) /CIGCP/GI(9) /CSL/CL(7) /CSV/CV(7)
     .       /CRPR/CR(3) /CTEVP/CT(8) /RFPR/RF(10) /METH/ M
C
C.... IF THE PROPERTIES OF OXYGEN ARE TO BE CALCULATED A CALL TO THIS
C.... SUBROUTINE MUST BE THE FIRST CALL STATEMENT IN THE MAIN PROGRAM
C
C.... THE COMMON BLOCKS INITIALIZED IN THIS ROUTINE HOLD THE FOLLOWING
C.... INFORMATION -
C
C         /CEOS/ G(41)   COEFFICIENTS OF THE EQUATION OF STATE
C
C         /CIGCP/ CIG(9) COEFFICIENTS OF THE IDEAL GAS HEAT CAPACITY EQUATION
C
C         /CVPN/ GV(11)  COEFFICIENTS OF THE VAPOR PRESSURE EQUATION
C
C         /CRPR/ CR(3)   THE CRITICAL PROPERTIES IN THE SAME UNITS AS THE
C                        EQUATION OF STATE
C                        CR(1)=CRITICAL PRESSURE
C                        CR(2)=CRITICAL DENSITY
C                        CR(3)=CRITICAL TEMPERATURE
C
C         /CSL/ SL(7)    COEFFICIENTS OF EQUATION USED TO APPROXIAMATE
C                        THE SATURATED LIQUID DENSITY AS A FUNCTION OF
C                        TEMPERATURE
C
C         /CSV/ SV(7)    COEFFICIENTS OF EQUATION USED TO APPROXIMATE
C                        THE SATURATED VAPOR DENSITY AS A FUNCTION
C                        OF THE TEMPERATURE
C
C         /RFPR/ RF(10)  REFERENCE PROPERTIES
C                        RF(1)=REFFERENCE ENTHALPY AT TEMPERATURE TOH
C                        RF(2)=REFFERENCE ENTROPY AT TEMPERATURE TOS
C                        RF(3)=TEMPERATURE TOH
C                        RF(4)=TEMPERATURE TOS
C                        RF(5)=GAS CONSTANT IN UNITS OF EQUATION OF STATE - R
C                        RF(6)=CONVERSION FACTOR TO CHANGE UNITS OF
C                              THE EQUATION OF STATE TO DESIRED ENERGY UNITS
C                        RF(7)=MOLECULAR WEIGHT
C                        RF(8)=TRIPLE POINT TEMPERATURE
C                        RF(9)  - NOT USED
C                        RF(10) - NOT USED
C
C         /METH/ M       INDICATES METHOD TO BE USED IN THE CALCULATION
C                        OF LIQUID PROPERTIES
C                        M=1  INDICATES ISOTHERM INTEGRATION THROUGH THE DOME
C                        M=2  INDICATES THE USE OF THE CLAPEYRON EQUATION
C                             THROUGH THE DOME
      M = 1
      G( 1) = -0.43090454E-02
      G( 2) =  0.35201737E+00
      G( 3) = -0.58362214E+01
      G( 4) =  0.24350909E+03
      G( 5) = -0.12463612E+05
      G( 6) =  0.12080882E-03
      G( 7) = -0.55031700E-01
      G( 8) = -0.10775786E-03
```

```
*********     DATA02     *********

   58          G( 9) =   0.27893571E+04
   59          G(10) =  -0.70406317E-05
   60          G(11) =   0.73426722E-02
   61          G(12) =  -0.59386982E+00
   62          G(13) =  -0.63616842E-04
   63          G(14) =   0.33034015E-03
   64          G(15) =  -0.85769299E-01
   65          G(16) =  -0.75461916E-05
   66          G(17) =   0.99643836E-07
   67          G(18) =   0.62468110E-04
   68          G(19) =  -0.73169850E-06
   69          G(20) =   0.11734852E+04
   70          G(21) =  -0.39878104E+06
   71          G(22) =  -0.16682114E+02
   72          G(23) =   0.10949860E+06
   73          G(24) =  -0.17843345E-01
   74          G(25) =   0.18752562E+01
   75          G(26) =  -0.94101786E-04
   76          G(27) =  -0.57539682E+00
   77          G(28) =  -0.20715572E-07
   78          G(29) =   0.15747135E-04
   79          G(30) =  -0.10757921E-09
   80          G(31) =  -0.70023860E-08
   81          G(32) =   0.34354651E-07
   82          G(41) =  -0.56000000E-02
   83          GV( 1) = -0.55819321E+03
   84          GV( 2) = -0.12118871E+03
   85          GV( 3) = -0.83456212E-01
   86          GV( 4) =  0.0
   87          GV( 5) =  0.16875024E-04
   88          GV( 6) = -0.21262477E-06
   89          GV( 7) =  0.95741097E-09
   90          GV( 8) = -0.16617640E-11
   91          GV( 9) =  0.27545606E+02
   92          GV(10) =  0.26603644E-02
   93          GV(11) =  0.19157553E+01
   94          GI( 1) = -0.49819985E+04
   95          GI( 2) =  0.23024778E+03
   96          GI( 3) = -0.34556532E+01
   97          GI( 4) =  0.35218767E+01
   98          GI( 5) = -0.43542022E-04
   99          GI( 6) =  0.13463534E-07
  100          GI( 7) =  0.16205983E-10
  101          GI( 8) =  0.10314685E+01
  102          GI( 9) =  0.22391811E+04
  103          CL( 1) =  0.23418919E+02
  104          CL( 2) =  0.63188646E+02
  105          CL( 3) = -0.17790822E+03
  106          CL( 4) =  0.41295084E+03
  107          CL( 5) = -0.50136455E+03
  108          CL( 6) =  0.24255815E+03
  109          CL( 7) =  0.11440740E+01
  110          CV( 1) =  0.18784168E+01
  111          CV( 2) = -0.96025698E+01
  112          CV( 3) =  0.21060675E+02
  113          CV( 4) = -0.88962992E+02
  114          CV( 5) =  0.14633955E+03
  115          CV( 6) = -0.12348431E+03
```

```
*********     DATA02     *********

              CV( 7) = -0.83291153E-01
              CR( 1) =  0.49770000E+02
              CR( 2) =  0.13630000E+02
              CR( 3) =  0.15458100E+03
              CT( 1) = -0.11152354E-02
              CT( 2) =  0.11160348E-01
              CT( 3) = -0.72432619E-04
              CT( 4) =  0.88579161E-05
              CT( 5) = -0.56777226E-06
              CT( 6) =  0.18657256E-07
              CT( 7) = -0.29899729E-09
              CT( 8) =  0.18530363E-11
              RF( 1) =  0.86820000E+04
              RF( 2) =  0.20503700E+03
              RF( 3) =  0.29815000E+03
              RF( 4) =  0.29815000E+03
              RF( 5) =  0.08205390E+00
              RF( 6) =  0.10132780E+03
              RF( 7) =  0.31998800E+02
              RF( 8) =  0.54351300E+02
              RETURN
              END
```

******** SUBRØUTINE DCALC

```
      SUBROUTINE DCALC(D,T,P,DL,DH)
      DATA MAX/30/
      DATA EPS/7.5E-3/
C
C.... ROUTINE TO PREFORM ITERATIVE SOLUTION OF THE EQUATION OF STATE
C
C.... DL IS LOWER BOUND ON DENSITY
C.... DH IS UPPER BOUND ON DENSITY
C
C.... THE DESIRED DENSITY MUST LIE BETWEEN DL AND DH
C
C.... ALGORITHM IS MODIFIED VERSION OF
C.... 'A QUADRATIC FORMULA FOR FINDING THE ROOT OF AN EQUATION'  BY
C.... LI. G. CHAMBERS MATHEMATICS OF COMPUTATION VOL 25 NO 114 APRIL (1971)
C
      IDS=0
      IC=0
    2 CONTINUE
      D1=DL
      D2=DH
      CALL PFND(T,D1,P1)
      CALL PFND(T,D2,P2)
      Y1=P1-P
      Y2=P2-P
    1 IC=IC+1
      IF(IC.GT.MAX)GO TO 5
      IF(IDS.EQ.1)GO TO 7
      IF(IC.GT.20)GO TO 6
      DS=(D1*Y2-D2*Y1)/(Y2-Y1)
      GO TO 8
    7 DS  = (D1+D2)/2.0
    8 CONTINUE
      CALL PFND(T,DS,PS)
      YS=PS-P
      D3=DS*Y1*Y2/((YS-Y2)*(YS-Y1)) +
     .   D1*YS*Y2/((Y1-Y2)*(Y1-YS)) +
     .   D2*YS*Y1/((Y2-Y1)*(Y2-YS))
      CALL PFND(T,D3,P3)
      Y3=P3-P
      IF(DABS(Y3).LE.EPS)GO TO 3
      IF(Y3.GT.0.0) GO TO 12
      IF(Y3.LT.Y1)GO TO 12
      Y1=Y3
      D1=D3
   12 IF(YS.GT.0.0) GO TO 13
      IF(YS.LT.Y1)GO TO 13
      Y1=YS
      D1=DS
   13 IF(Y3.LT.0.0) GO TO 14
      IF(Y3.GT.Y2)GO TO 14
      Y2=Y3
      D2=D3
   14 IF(YS.LT.0.0) GO TO 1
      IF(YS.GT.Y2)GO TO 1
      Y2=YS
      D2=DS
      GO TO 1
```

```
*********     DCALC     *********

          3 CONTINUE
            D=D3
            RETURN
          5 WRITE(6,300)T,P,DL,DH,D1,Y1,D2,Y2,DS,YS,D3,Y3
        300 FORMAT(' *** DCALC FAILED TO CONVERGE ***',/,
           .     '         T  = ',G15.7,/,
           .     '         P  = ',G15.7,/,
           .     '         DL = ',G15.7,/,
           .     '         DH = ',G15.7,/,
           .     '         D1 = ',G15.7,/,
           .     '         Y1 = ',G15.7,/,
           .     '         D2 = ',G15.7,/,
           .     '         Y2 = ',G15.7,/,
           .     '         DS = ',G15.7,/,
           .     '         YS = ',G15.7,/,
           .     '         D3 = ',G15.7,/,
           .     '         Y3 = ',G15.7)
            D=DS
            RETURN
          6 CONTINUE
            IDS=1
            IC=0
            GO TO 2
            END
```

******** SUBRØUTINE DENSØN

```
      SUBROUTINE DENSON(TEMP,PRES,NGAS,DENS,ZEE)
C
      COMMON /METH/ M
C
C     INPUT TO THIS SUBROUTINE MUST BE IN BRITISH UNITS
C
      T = TEMP
      P = PRES
      M = 1
C
      IF(NGAS.EQ.1) KF = 1
      IF(NGAS.EQ.18) KF = 2
C
C     KF = 1  CALL IN OXYGEN PARAMETERS
C     KF = 2  CALL IN NITROGEN PARAMETERS
C
      IF(KF.EQ.1) CALL DATAO2
      IF(KF.EQ.2) CALL DATAN2
C
      CALL DFNDB(T,P,D,Z,D)
      DENS = D
      ZEE = Z
C
      RETURN
      END
```

********* SUBRØUTINE DFND

```
      SUBROUTINE DFND(T,P,D,Z,K)
      COMMON /RFPR/ RF(10)
      COMMON /CRPR/CP(3)
C
C.... ROUTINE TO GENERATE TRAIL DENSITIES FOR ITERATIVE SOLUTION OF
C.... THE EQUATION OF STATE FOR DENSITY GIVEN TEMPERATURE AND PRESSURE
C
C     K =0  INDICATES SINGLE PHASE POINT
C     K =1  INDICATES T + P ARE FOR THE SATURATED LIQUID
C     K =2  INDICATES T + P ARE FOR THE SATURATED VAPOR
C
      IF((K.LE.2).AND.(K.GE.0))GO TO 1
      WRITE(6,300)K
  300 FORMAT(' *** ERROR IN CALL DFND ***',/,
     .       '      K MUST EQUAL 0, 1, OR 2',/,
     .       '      K = ',I10)
      RETURN
    1 PC=CP(1)
      DC=CR(2)
      TC=CR(3)
      R = RF(5)
      IF(K.GT.0)GO TO 5
      IF(T.GE.TC)GO TO 2
      VP=VPN(T)
      IF(P.LE.VP)GO TO 3
    4 DH  = 3.1 * DC
      DL=DSATL(T)
      CALL DCALC(D,T,P,DL,DH)
      Z = P/(D*R*T)
      RETURN
    3 DL  = 0.0
      DH=DSATV(T)
      CALL DCALC(D,T,P,DL,DH)
      Z = P/(D*R*T)
      RETURN
    2 DL  = 0.0
      DH  = 3.1 * DC
      IF((T.GT.1000.0).AND.(P.LT.300.0)) DH=DC
      CALL DCALC(D,T,P,DL,DH)
      Z = P/(D*R*T)
      RETURN
    5 IF(K.EQ.1)GO TO 4
      GO TO 3
      END
```

SUBRØUTINE DFNDB ***

```
      SUBROUTINE DFNDB(TB,PB,DB,ZB,K)
      COMMON /RFPR/ RF(10)
      WT=RF(7)
      T   = TB/1.8
      P   = PB * 6.8947572E+3/1.01325E+5
      CALL DFND(T,P,D,Z,K)
      DB  = D * WT * 2.8316847E-2/453.59237E-3
      ZB = Z
      RETURN
      END
```

******** FUNCTIØN DIAG

```
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C                    * ROUTINE NAME = DIAGNOSTIC TRACE ROUTINE       *
C                    * ROUTINE LANG = FORTRAN V UNIVAC 1108 EXEC 2   *
C                    * PROGRAMMER   = R. BOLLINGER 1943 102 26933    *
C                    * DATE CODED   = APRIL 6,1970                   *
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION DIAG(I,NAME)
C
C         ***** EXPLANATION OF THE CALLING SEQUENCE
C               I     - PRINT FLAG
C                       IF I = -2, THE DIAGNOSTIC TRACE WILL BE TURNED OFF.
C                       IF I = -1, THE DIAGNOSTIC TRACE WILL BE TURNED ON.
C                       IF I =  0, DIAG WILL PRINT THE XXXXXX ENTRED MESSAGE.
C                       IF I =  1, DIAG WILL PRINT THE XXXXXX EXITED MESSAGE.
C                       IF I =  2, DIAG WILL PRINT THE XXXXXX INTRNL MESSAGE.
C
C               NAME  - THE NAME OF THE ROUTINE TO BE PRINTED OUT WITH
C                       I = 0,1,2.
C
      LOGICAL PAGE,DIAG
C
      DIMENSION MSG(3)
C
      DATA (MSG(J),J=1,3)/'ENTRED','EXITED','INTRNL'/
C
      DIAG = .FALSE.
      IF(I)10,40,40
   10 IF(I + 1) 20,30,30
   20 IFLG = 0
      RETURN
   30 IFLG = 1
      IF(PAGE(0)) WRITE(6,6000)
      RETURN
   40 IF(IFLG.EQ.0) RETURN
      WRITE(6,6010) NAME,MSG(I+1)
      DIAG = .TRUE.
      RETURN
 6000 FORMAT('0**DIAGNOSTIC TRACE**')
C
 6010 FORMAT(1XA6,1XA6)
      END
```

********* FUNCTIØN DPDD

```
      FUNCTION DPDD(T,D)
      COMMON /RFPR/ RF(10)
      COMMON /CEOS/G(41)
      COMMON /SCRH/ X(40)
C
C.... CALCULATES DP/DD OF THE EQUATION OF STATE
C
      R=RF(5)
      D2=D*D
      D3=D2*D
      D4=D3*D
      D5=D4*D
      D6=D5*D
      D7=D6*D
      D8=D7*D
      D9=D8*D
      D10=D9*D
      D11=D10*D
      D12=D11*D
      D13=D12*D
      TS= SQRT(T)
      T2=T*T
      T3=T2*T
      T4=T3*T
      GM=G(41)
      F= EXP(GM*D2)
      F1=2.00*F*GM*D
      F21=3.000*F*D2 +F1*D3
      F22=5.000*F*D4 +F1*D5
      F23=7.000*F*D6 +F1*D7
      F24=9.000*F*D8 +F1*D9
      F25=11.00*F*D10+F1*D11
      F26=13.00*F*D12+F1*D13
      X( 1)=2.00*D*T
      X( 2)=2.00*D*TS
      X( 3)=2.00*D
      X( 4)=2.00*D/T
      X( 5)=2.00*D/T2
      X( 6)=3.00*D2*T
      X( 7)=3.00*D2
      X( 8)=3.00*D2/T
      X( 9)=3.00*D2/T2
      X(10)=4.00*D3*T
      X(11)=4.00*D3
      X(12)=4.00*D3/T
      X(13)=5.00*D4
      X(14)=6.00*D5/T
      X(15)=6.00*D5/T2
      X(16)=7.00*D6/T
      X(17)=8.00*D7/T
      X(18)=8.00*D7/T2
      X(19)=9.00*D8/T2
      X(20)=F21/T2
      X(21)=F21/T3
      X(22)=F22/T2
      X(23)=F22/T4
      X(24)=F23/T2
```

```
********    DPDD    ********

          X(25)=F23/T3
          X(26)=F24/T2
          X(27)=F24/T4
          X(28)=F25/T2
          X(29)=F25/T3
          X(30)=F26/T2
          X(31)=F26/T3
          X(32)=F26/T4
          N=32
          DPDD=0.0
          DO 1 K=1,N
        1 DPDD=DPDD+X(K)*G(K)
          DPDD=DPDD+R*T
          RETURN
          END
```

********* FUNCTIØN DPDDB

```
      FUNCTION DPDDB(TB,DB)
      COMMON /RFPR/ RF(10)
      WT=RF(7)
      T   = TB/1.8
      D   = DB * 453.59237E-3/(WT * 2.8316847E-2)
      DPDDB = DPDD(T,D) * 453.59237E-3 * 1.01325E+5/(WT * 6.8947572E+3 *
     1 2.8316847E-2)
      RETURN
      END
```

******** FUNCTIØN DPDT

```
      FUNCTION DPDT(T,D)
      COMMON /CEOS/G(41)
      COMMON /RFPR/ RF(10)
      COMMON /SCRH/ X(40)
C
C.... CALCULATES DP/DT OF THE EQUATION OF STATE
C
      R=RF(5)
      D2=D*D
      D3=D2*D
      D4=D3*D
      D5=D4*D
      D6=D5*D
      D7=D6*D
      D8=D7*D
      D9=D8*D
      D10=D9*D
      D11=D10*D
      D12=D11*D
      D13=D12*D
      TS= SQRT(T)
      T2=T*T
      T3=T2*T
      T4=T3*T
      T5=T4*T
      GM=G(41)
      F= EXP(GM*D2)
      X( 1)=D2
      X( 2)=D2/(2.00*TS)
      X( 3)=0.0
      X( 4)=-D2/T2
      X( 5)=-2.00*D2/T3
      X( 6)=D3
      X( 7)=0.0
      X( 8)=-D3/T2
      X( 9)=-2.00*D3/T3
      X(10)=D4
      X(11)=0.0
      X(12)=-D4/T2
      X(13)=0.0
      X(14)=-D6/T2
      X(15)=-2.00*D6/T3
      X(16)=-D7/T2
      X(17)=-D8/T2
      X(18)=-2.00*D8/T3
      X(19)=-2.00*D9/T3
      X(20)=-2.00*D3*F/T3
      X(21)=-3.00*D3*F/T4
      X(22)=-2.00*D5*F/T3
      X(23)=-4.00*D5*F/T5
      X(24)=-2.00*D7*F/T3
      X(25)=-3.00*D7*F/T4
      X(26)=-2.00*D9*F/T3
      X(27)=-4.00*D9*F/T5
      X(28)=-2.00*D11*F/T3
      X(29)=-3.00*D11*F/T4
      X(30)=-2.00*D13*F/T3
```

```
*********     DPDT    :*********

          X(31)=-3.00*D13*F/T4
          X(32)=-4.00*D13*F/T5
          N=32
          DPDT=0.0
          DO 1 K=1,N
          DPDT=DPDT+X(K)*G(K)
          DPDT=DPDT+R*D
          RETURN
          END
```

********* FUNCTIØN DPDTB

```
      FUNCTION DPDTB(TB,DB)
      COMMON /RFPR/ RF(10)
      WT=RF(7)
      T   = TB/1.8
      D   = DB * 453.59237E-3/(WT * 2.8316847E-2)
      DPDTB = DPDT(T,D) * 1.01325E+5/(1.8 * 6.8947572E+3)
      RETURN
      END
```

********* SUBRØUTINE DPDTVP

```
      SUBROUTINE DPDTVP(T,P,DPDT)
      COMMON/CVPN/G(11) /CRPR/CR(3)
      COMMON /SCRH/ X(40)
C
C.... CALCULATE DP/DT FOR THE VAPOR PRESSURE EQUATION
C
      TC=CR(3)
      A=G(11)
      T2=T*T
      T3=T*T2
      T4=T*T3
      T5=T*T4
      X(1) = -1.0/T2
      X(2) = 0.0
      X(3) = 1.0
      X(4) = 2.0*T
      X(5) = 3.0*T2
      X(6) = 4.0*T3
      X(7) = 5.0*T4
      X(8) = 6.0*T5
      X(9) = 1.0/T
      X(10) = (TC-T)**(A-1.0)*(-A)
      DPDT = 0.0
      DO 1 I=1,10
    1 DPDT=DPDT+X(I)*G(I)
      DPDT=DPDT*P
      RETURN
      END
```

******** FUNCTIØN DSATL

```
      FUNCTION DSATL(T)
      COMMON /CRPR/CR(3) /CSL/G(7)
      COMMON /SCRH/ B(40)
C
C.... THIS FUNCTION SUPPLIES AN APPROXIMATE VALUE FOR THE
C.... DENSITY OF THE SATURATED LIQUID
C
      TC=CR(3)
      X=(TC-T)/TC
      X2=X*X
      X3=X*X2
      X4=X*X3
      X5=X*X4
      B(1) = 1.0
      B(2)=X
      B(3)=X2
      B(4)=X3
      B(5)=X4
      B(6)=X5
      B(7) = ALOG(X)
      DSL = 0.0
      DO 1 I=1,7
    1 DSL=DSL+B(I)*G(I)
      DSATL=DSL
      RETURN
      END
```

********* FUNCTIØN DSATV

```
      FUNCTION DSATV(T)
      COMMON /CRPR/CR(3) /CSV/G(7)
      COMMON /SCRH/ B(40)
C
C.... THIS FUNCTION SUPPLIES AN APPROXIMATE VALUE FOR THE
C.... DENSITY OF THE SATUREATED VAPOR
C
      TC=CR(3)
      X=(TC-T)/TC
      X2=X*X
      X3=X*X2
      X4=X*X3
      X5=X*X4
      B(1) = 1.0
      B(2)=X
      B(3)=X2
      B(4)=X3
      B(5)=X4
      B(6)=X5
      B(7) = ALOG(X)
      DSV = 0.0
      DO 1 I=1,7
    1 DSV=DSV+B(I)*G(I)
      DSV = EXP(DSV)
      DSATV=DSV
      RETURN
      END
```

********   PRØCEDURE DEFINITIØN PRØCESSØR - DUMMY

```
DUMMY*  PROC
C
      DIMENSION  SCRTCH(100)
C
      COMMON /CDUMMY/ IGRBAG(100)
C
      EQUIVALENCE  (IGRBAG,SCRTCH)
C
C         ***** CDUMMY VARIABLE DESCRIPTION
C
C               IGRBAG - A SCRATCH AREA FOR VARIABLE STORAGE WHICH IS
C                        NOT TO BE SAVE FROM ON ROUTINE TO ANOTHER.
 END
```

```
      SUBROUTINE ECLSS
C
      LOGICAL  JP,PAGE
C
C
          INCLUDE CACCUM
          INCLUDE CAPU
          INCLUDE CDCYCL
          INCLUDE CECLSS
          INCLUDE CENG
          INCLUDE CFUEL
          INCLUDE CHEX
          INCLUDE CHSORC
          INCLUDE CIOUNT
          INCLUDE CMATRL
          INCLUDE CONST
          INCLUDE CPUMP
          INCLUDE CTANK
          INCLUDE TABLOK
C
      DIMENSION WDOTX(MHX,2)
      EQUIVALENCE (WDOTCF,WDOTX)
C
C  ******************************************************************
C
      JP = PAGE(0)
C
      WRITE (IOT,6000)
 6000 FORMAT(//T38,'*** INITIATE PROGRAM AND CHARACTERIZE LIFE SUPPORT P
     1ARAMETERS ***'//)
      WRITE (IOT,6001)
 6001 FORMAT(          //T2'PARAMETER',T14,'CYCLE-1',T26,'CYCLE-2',
     1 T38,'CYCLE-3',T50,'CYCLE-4',T62,'CYCLE-5',T74,'CYCLE-6',T86,
     2 'CYCLE-7',T98,'CYCLE-8',T110,'CYCLE-9',T121,'CYCLE-10'/)
C
C
C  ******************************************************************
C
C     BEGIN COMPUTATIONS FOR SUPERCRITICAL STORAGE
C
C  ******************************************************************
C
C         *** SET SELECTED INPUT PARAMETERS FOR CONVENIENCE
C
      POPO2 = SOPRES(1,1)
      POPN2 = SOPRES(2,1)
      PVPO2 = SVPRES(1,1)
      PVPN2 = SVPRES(2,1)
      TO2IN = SITEMP(1,1)
      TN2IN = SITEMP(2,1)
C
C         *** COMPUTE QUANTITY OF FLUIDS CONSUMED FOR LIFE SUPPORT,
C         *** LEAKAGE, AND AIRLOCK OR CABIN REPRESSURIZATION.COMPUTE
C         *** GAS REQD. FOR REPRESSURIZATION AND TOTAL GAS REQMTS.
C
      O2MCON = 0.0
      O2LCON = 0.0
```

```
********    ECLSS    ********

            N2LCON = 0.0
            I = 0
            DO 10 II = 1,NDCYCL,2
            I = I + 1
            O2MWT(I) = (O2FNOM/24.0) * NCREW * DCYCLE(II)
            O2MCON   = O2MCON + O2MWT(I)                          8 LBS
            O2LWT(I) = 0.21 * (GLKRAT/24.0) * DCYCLE(II)
            O2LCON   = O2LCON + O2LWT(I)                          8 LBS
            N2LWT(I) = 0.79 * (GLKRAT/24.0) * DCYCLE(II)
            N2LCON   = N2LCON + N2LWT(I)                          8 LBS
      C
         10 CONTINUE
      C
            GASWGT = CABVOL/13.2743                  8 AIR AT 14.7 PSIA AND 70 F
            O2REPR = 0.21 * NRPRES * GASWGT
            N2REPR = 0.79 * NRPRES * GASWGT
            O2CONS = O2MCON + O2LCON + O2REPR                     8  LBS
            N2CONS = N2LCON + N2REPR                              8  LBS
      C
      C     ******************************************************************
      C
      C         *** OUTPUT THE DATA COMPUTED TO THIS POINT ***
      C
            WRITE (IOT,6002)
       6002 FORMAT(/T3,'COMPUTE QUANTITY OF FLUIDS CONSUMED EACH INTERVAL AND
           1FOR ENTIRE MISSION')
            WRITE (IOT,600)
        600 FORMAT(/)
            WRITE (IOT,601) (O2MWT(N),N=1,KCYCLE)
        601 FORMAT(T3,'O2MWT  =',T13,10(F8.5,4X))
            WRITE (IOT,602) (O2LWT(N),N=1,KCYCLE)
        602 FORMAT(T3,'O2LWT  =',T13,10(F8.5,4X))
            WRITE (IOT,603) (N2LWT(N),N=1,KCYCLE)
        603 FORMAT(T3,'N2LWT  =',T13,10(F8.5,4X))
            WRITE (IOT,604) O2MCON, O2LCON
        604 FORMAT(T3,'O2MCON =',T13,F8.3,T25,'O2LCON =',T37,F8.3)
            WRITE (IOT,605) N2LCON, GASWGT
        605 FORMAT(T3,'N2LCON =',T13,F8.3,T25,'GASWGT =',T37,F8.3)
            WRITE (IOT,606) O2REPR, N2REPR
        606 FORMAT(T3,'O2REPR =',T13,F8.3,T25,'N2REPR =',T37,F8.3)
            WRITE (IOT,607) O2CONS, N2CONS
        607 FORMAT(T3,'O2CONS =',T13,F8.3,T25,'N2CONS =',T37,F8.3)
      C
      C         *** COMPUTE THE CONTINGENCY RESERVE GASES REQD.FOR THE
      C         *** MISSION
      C
            O2MRES = (O2FNOM/24.0) * NCREW * NDARES * 24.0
            O2LRES = 0.21 * (GLKRAT/24.0) * NDARES * 24.0
            N2LRES = 0.79 * (GLKRAT/24.0) * NDARES * 24.0
            O2RES  = O2MRES + O2LRES                              8 LBS
            N2RES  = N2LRES                                       8 LBS
      C
      C         *** COMPUTE USEABLE GAS CONSUMABLES TOTAL
      C
            O2TOTU = O2CONS + O2RES                               8  LBS
            N2TOTU = N2CONS + N2RES                               8  LBS
      C
```

```
C   ********************************************************************
C
C         *** OUTPUT THE DATA COMPUTED TO THIS POINT ***
C
      WRITE (IOT,6003)
 6003 FORMAT(/T3,'CONTINGENCY RESERVE FLUIDS REQD.FOR MISSION')
      WRITE (IOT,600)
      WRITE (IOT,608) O2MRES, O2LRES, N2LRES
  608 FORMAT(T3,'O2MRES =',T13,F8.3,T25,'O2LRES =',T37,F8.3,T50,'N2LRES
     1=',T60,F8.3)
      WRITE (IOT,609) O2RES, N2RES
  609 FORMAT(T3,'O2RES  =',T13,F8.3,T25,'N2RES  =',T37,F8.3)
C
C
      WRITE (IOT,6004)
 6004 FORMAT(/T3,'TOTAL USABLE FLUIDS AVAILABLE')
      WRITE (IOT,600)
      WRITE (IOT,610) O2TOTU, N2TOTU
  610 FORMAT(T3,'O2TOTU =',T13,F8.3,T25,'N2TOTU =',T37,F8.3)
C
C         *** COMPUTE NOMINAL FLOWRATE AND REPRESSURIZATION FLOWRATE
C         *** FOR GASES, COMPUTE MAX FLOW RATE FOR GASES.
C         *** COMPUTE QUANTITY GASES CONSUMED EACH INTERVAL
C
C
      I = 0
      DO 15 II = 1,NDCYCL,2
      I = I + 1
      WDOTON(I) = (O2MWT(I) + O2LWT(I))/DCYCLE(II)       @ LBS PER HR
      WDOTNN(I) = N2LWT(I)/DCYCLE(II)                    @ LBS PER HR
      IF(RPRTIM(I).EQ.0.0) GO TO 18
      WDOTOR(I) = (O2REPR/NRPRES)/RPRTIM(I)              @ LBS PER HR
      WDOTNR(I) = (N2REPR/NRPRES)/RPRTIM(I)              @ LBS PER HR
   18 WDTO2(I) = WDOTON(I) + WDOTOR(I)
      WDTN2(I)  = WDOTNN(I) + WDOTNR(I)                  @ LBS PER HR
      WTO2(I)   = WDOTON(I)*DCYCLE(II) + WDOTOR(I)*RPRTIM(I)      @ LBS
      WTN2(I)   = WDOTNN(I)*DCYCLE(II) + WDOTNR(I)*RPRTIM(I)      @ LBS
   15 CONTINUE
C
      WDTOMX = 0.0
      WDTNMX = 0.0
      DO 17 I = 1,KCYCLE
      IF(WDTO2(I).LT.WDTOMX) GO TO 16
      WDTOMX = WDTO2(I)
      IMAX = I
   16 IF(WDTN2(I).LT.WDTNMX) GO TO 17
      WDTNMX = WDTN2(I)
      JMAX = I
   17 CONTINUE
C
      WDOTI(1) = WDTOMX/3600.0                   @ LBS PER SECOND
      WDOTI(2) = WDTNMX/3600.0                   @ LBS PER SECOND
      WDOTT(1) = WDOTI(1)                        @ LBS PER SECOND
      WDOTT(2) = WDOTI(2)                        @ LBS PER SECOND
C
C     ******************************************************************
C
C         *** OUTPUT THE DATA COMPUTED TO THIS POINT ***
```

```
C
      WRITE (IOT,6005)
 6005 FORMAT(/T3,'NOMINAL FLOWRATE, REPRESSURIZATION FLOWRATE, MAX-FLOWR
     1ATE, AND QUANTITY EACH FLUID CONSUMED PER INTERVAL')
      WRITE (IOT,600)
      WRITE (IOT,611) (WDOTON(N),N=1,KCYCLE)
  611 FORMAT(T3,'WDOTON =',T13,10(F8.5,4X))
      WRITE (IOT,612) (WDOTNN(N),N=1,KCYCLE)
  612 FORMAT(T3,'WDOTNN =',T13,10(F8.5,4X))
      WRITE (IOT,613) (WDOTOR(N),N=1,KCYCLE)
  613 FORMAT(T3,'WDOTOR =',T13,10(F8.5,4X))
      WRITE (IOT,614) (WDOTNR(N),N=1,KCYCLE)
  614 FORMAT(T3,'WDOTNR =',T13,10(F8.5,4X))
      WRITE (IOT,615)  (WDTO2(N),N=1,KCYCLE)
  615 FORMAT(T3,'WDTO2  =',T13,10(F8.5,4X))
      WRITE (IOT,616)  (WDTN2(N),N=1,KCYCLE)
  616 FORMAT(T3,'WDTN2  =',T13,10(F8.5,4X))
      WRITE (IOT,617) WDTOMX, WDTNMX
  617 FORMAT(T3,'WDTOMX =',T13,F8.3,T25,'WDTNMX =',T37,F8.3)
      WRITE (IOT,619) (WTO2(N),N=1,KCYCLE)
  619 FORMAT(T3,'WTO2   =',T13,10(F8.4,4X))
      WRITE (IOT,620) (WTN2(N),N=1,KCYCLE)
  620 FORMAT(T3,'WTN2   =',T13,10(F8.4,4X))
C
C         *** DETERMINE INITIAL FLUID TANK TEMPERATURES
C
      CALL FINTAB(NTBID(8))
      XTAB(1) = POPO2
      XTAB(2) = RHOBEG(1)
      TEMPO2  = MIPE(2,XTAB)
C
      CALL FINTAB(NTBID(41))
      XTAB(1) = POPN2
      XTAB(2) = RHOBEG(2)
      TEMPN2  = MIPE(2,XTAB)
C
C         *** DETERMINE INITIAL CSUBV VALUES FOR TANK T AND P CONDITIONS
C
      CISBVO = CSUBV(TEMPO2,POPO2,1)
      CISBVN = CSUBV(TEMPN2,POPN2,18)
C
C         *** COMPUTE FLUID COMPRESSIBILITY AT FINAL TANK CONDITIONS
C
      ZFO  =  ZGET(TKFTEM(1),TKFPRS(1),1)
      ZFN  =  ZGET(TKFTEM(2),TKFPRS(2),18)
C
C         *** COMPUTE THE PERCENT OF USABLE FLUIDS WITHDRAWN
C         *** EACH INTERVAL
C
      TKOW = 0.0
      TKNW = 0.0
      QDTOMX = 0.0
      QDTNMX = 0.0
      HWTOMX = 0.0
      HWTNMX = 0.0
      HWTOTT = 0.0
      HWTNTT = 0.0
      TWTOMX = 0.0
```

```
      TWTNMX = 0.0
      TWTOTT = 0.0
      TWTNTT = 0.0
C
      DO 20 I = 1,KCYCLE
      TKOW = TKOW + WTO2(I)
      TKO2DP(I) = TKOW
      TKNW = TKNW + WTN2(I)
      TKN2DP(I) = TKNW
C
      PCOXWD(I) = TKO2DP(I)/O2TOTU
      PCN2WD(I) = TKN2DP(I)/N2TOTU
C
C        *** COMPUTE DENSITY OF FLUIDS AS FUNCTION OF PERCENT WITHDRAWN
C
      C1 = 144.0/(RHOBEG(1) * (1544.2546/31.9988))
      C2 = 144.0/(RHOBEG(2) * (1544.2546/28.0134))
C
      C3 = 1.0 - ((C1 * TKFPRS(1))/(ZFO * TKFTEM(1)))
      C4 = 1.0 - ((C2 * TKFPRS(2))/(ZFN * TKFTEM(2)))
C
      O2RHO(I) = RHOBEG(1) * (1.0 - (PCOXWD(I) * C3))
      N2RHO(I) = RHOBEG(2) * (1.0 - (PCN2WD(I) * C4))
C
C        *** COMPUTE FLUID TEMPERATURE IN TANKS FOR EACH INTERVAL
C
      CALL FINTAB (NTBID(8))
      XTAB(1) = POPO2
      XTAB(2) = O2RHO(I)
      O2TEMP(I) = MIPE(2,XTAB)
C
      CALL FINTAB (NTBID(41))
      XTAB(1) = POPN2
      XTAB(2) = N2RHO(I)
      N2TEMP(I) = MIPE(2,XTAB)
C
C        *** COMPUTE SPECIFIC HEAT INPUT (DQ/DM) FOR FLUIDS AS F(D,P)
C        *** COMPUTE ENERGY DERIVATIVE (DP/DU) FOR FLUIDS AS F(D,P)
C
      CALL PHTHON(O2TEMP(I),O2RHO(I),1,PHI,THETA)
      DQDMO2(I) = THETA
      DPDUO2(I) = PHI
C
      CALL PHTHON(N2TEMP(I),N2RHO(I),18,PHI,THETA)
      DQDMN2(I) = THETA
      DPDUN2(I) = PHI
C
C        *** SIZE    CONDITIONING HEAT EXCHANGERS FOR FLUIDS
C
      HLSO = OXENTH(POPO2,TLSNOM(1))
      HLSN = NIENTH(POPN2,TLSNOM(2))
      O2H(I) = OXENTH(POPO2,O2TEMP(I))
      N2H(I) = NIENTH(POPN2,N2TEMP(I))
C
      QDTOR(I) = WDTO2(I) * (HLSO - O2H(I))
      QDTOMX = AMAX1(QDTOMX,QDTOR(I))
      QDTNR(I) = WDTN2(I) * (HLSN - N2H(I))
      QDTNMX = AMAX1(QDTNMX,QDTNR(I))
```

```
********     ECLSS     ********

      C
      C          *** COMPUTE POWER REQUIRED TO PROVIDE ENERGY IN HEAT XCHNGR.
      C
            HWATO2(I) =  QDTOR(I) * 0.293
            HWTOMX = AMAX1(HWTOMX,HWATO2(I))
            HWTOTT = HWTOTT + HWATO2(I)
            HWATN2(I) =  QDTNR(I) * 0.293
            HWTNMX = AMAX1(HWTNMX,HWATN2(I))
            HWTNTT = HWTNTT + HWATN2(I)
      C
      C          *** SIZE O2 TANK AND N2 TANK HEAT REQUIREMENTS
      C
            QDTTKO(I)  = WDTO2(I) * DQDMO2(I)
            QDTTKN(I)  = WDTN2(I) * DQDMN2(I)
      C
      C     CALCULATE THE TANK HEATER RATINGS FOR EACH FLUID TANK BASED UPON
      C     INPUT HEATER DIAMETER AND LENGTH
      C     * PROGRAM WILL USE CALCULATED VALUES ONLY IF INPUT HTRFLX(2)=0.0 *
      C
            HTRRA1 = (TWTOMX/0.293)/(PI * HTRDIA(1) * HTRLNG(1))
            HTRRA2 = (TWTNMX/0.293)/(PI * HTRDIA(2) * HTRLNG(2))
      C
      C          *** COMPUTE POWER REQUIRED TO PROVIDE ENERGY INTO TANKS
      C
            TWATO2(I)  = QDTTKO(I) * 0.293
            TWTOMX = AMAX1(TWTOMX,TWATO2(I))
            TWTOTT = TWTOTT + TWATO2(I)
            TWATN2(I)  = QDTTKN(I) * 0.293
            TWTNMX = AMAX1(TWTNMX,TWATN2(I))
            TWTNTT = TWTNTT + TWATN2(I)
         20 CONTINUE
      C
            TOTWMX = HWTOMX + HWTNMX + TWTOMX + TWTNMX
      C
            TOTWAT = HWTOTT + HWTNTT + TWTOTT + TWTNTT
      C
            TOTPOW = TOTWAT/1000.0
      C
      C     ********************************************************************
      C
      C          *** OUTPUT THE DATA COMPUTED TO THIS POINT ***
      C
            JP = PAGE(0)
            WRITE (IOT,6116)
       6116 FORMAT(//T42,'*** CONTINUE COMPUTATION OF - ECLSS - PARAMETERS ***
           1'//)
            WRITE (IOT,6001)
            WRITE (IOT,6006)
       6006 FORMAT(/T3,'DETERMINE FLUID TANK CONDITIONS FOR DUTY CYCLE')
            WRITE (IOT,600)
            WRITE (IOT,621) TEMPO2, TEMPN2
        621 FORMAT(T3,'TEMPO2 =',T13,F8.2,T25,'TEMPN2 =',T37,F8.2)
            WRITE (IOT,622) CISBVO, CISBVN, ZFO, ZFN
        622 FORMAT(T3,'CISBVO =',T13,F8.4,T25,'CISBVN =',T37,F8.4,T50,'ZFO
           1=',T63,F8.4,T73,'ZFN    =',T86,F8.4)
            WRITE (IOT,623) (PCOXWD(N),N=1,KCYCLE)
        623 FORMAT(T3,'PCOXWD =',T13,10(F8.6,4X))
            WRITE (IOT,624) (PCN2WD(N),N=1,KCYCLE)
```

```
      624 FORMAT(T3,'PCN2WD =',T13,10(F8.6,4X))
          WRITE (IOT,625) (O2RHO(N),N=1,KCYCLE)
      625 FORMAT(T3,'O2RHO  =',T13,10(F8.4,4X))
          WRITE (IOT,626) (N2RHO(N),N=1,KCYCLE)
      626 FORMAT(T3,'N2RHO  =',T13,10(F8.4,4X))
          WRITE (IOT,627) (O2TEMP(N),N=1,KCYCLE)
      627 FORMAT(T3,'O2TEMP =',T13,10(F8.2,4X))
          WRITE (IOT,628) (N2TEMP(N),N=1,KCYCLE)
      628 FORMAT(T3,'N2TEMP =',T13,10(F8.2,4X))
          WRITE (IOT,629) (DQDMO2(N),N=1,KCYCLE)
      629 FORMAT(T3,'DQDMO2 =',T13,10(F8.2,4X))
          WRITE (IOT,630) (DQDMN2(N),N=1,KCYCLE)
      630 FORMAT(T3,'DQDMN2 =',T13,10(F8.2,4X))
          WRITE (IOT,631) (DPDUO2(N),N=1,KCYCLE)
      631 FORMAT(T3,'DPDUO2 =',T13,10(F8.3,4X))
          WRITE (IOT,632) (DPDUN2(N),N=1,KCYCLE)
      632 FORMAT(T3,'DPDUN2 =',T13,10(F8.3,4X))
C
          WRITE (IOT,6007)
     6007 FORMAT(/T3,'SIZE FLUID CONDITIONING HEAT EXCHANGERS')
          WRITE (IOT,600)
          WRITE (IOT,618) HLSO, HLSN
      618 FORMAT(T3,'HLSO   =',T13,F8.3,T25,'HLSN   =',T37,F8.3)
          WRITE (IOT,633) (O2H(N),N=1,KCYCLE)
      633 FORMAT(T3,'O2H    =',T13,10(F8.3,4X))
          WRITE (IOT,634) (N2H(N),N=1,KCYCLE)
      634 FORMAT(T3,'N2H    =',T13,10(F8.3,4X))
          WRITE (IOT,635) (QDTOR(N),N=1,KCYCLE)
      635 FORMAT(T3,'QDTOR  =',T13,10(F8.1,4X))
          WRITE (IOT,636) (QDTNR(N),N=1,KCYCLE)
      636 FORMAT(T3,'QDTNR  =',T13,10(F8.1,4X))
          WRITE (IOT,637) QDTOMX, QDTNMX
      637 FORMAT(T3,'QDTOMX =',T13,F8.1,T25,'QDTNMX =',T37,F8.1)
          WRITE (IOT,638) (HWATO2(N),N=1,KCYCLE)
      638 FORMAT(T3,'HWATO2 =',T13,10(F8.2,4X))
          WRITE (IOT,639) (HWATN2(N),N=1,KCYCLE)
      639 FORMAT(T3,'HWATN2 =',T13,10(F8.2,4X))
          WRITE (IOT,640) HWTOMX, HWTNMX
      640 FORMAT(T3,'HWTOMX =',T13,F8.2,T25,'HWTNMX =',T37,F8.2)
          WRITE (IOT,641) HWTOTT, HWTNTT
      641 FORMAT(T3,'HWTOTT =',T13,F8.1,T25,'HWTNTT =',T37,F8.1)
          WRITE (IOT,6008)
     6008 FORMAT(/T3,'SIZE FLUID TANK HEAT REQUIREMENTS')
          WRITE (IOT,600)
          WRITE (IOT,642) (QDTTKO(N),N=1,KCYCLE)
      642 FORMAT(T3,'QDTTKO =',T13,10(F8.1,4X))
          WRITE (IOT,643) (QDTTKN(N),N=1,KCYCLE)
      643 FORMAT(T3,'QDTTKN =',T13,10(F8.1,4X))
          WRITE (IOT,644) (TWATO2(N),N=1,KCYCLE)
      644 FORMAT(T3,'TWATO2 =',T13,10(F8.1,4X))
          WRITE (IOT,645) (TWATN2(N),N=1,KCYCLE)
      645 FORMAT(T3,'TWATN2 =',T13,10(F8.1,4X))
          WRITE (IOT,646) TWTOMX, TWTNMX
      646 FORMAT(T3,'TWTOMX =',T13,F8.1,T25,'TWTNMX =',T37,F8.1)
          WRITE (IOT,670) HTRRA1, HTRRA2
      670 FORMAT(T3,'O2 TANK HEATER RATING=',F6.1,T35,'N2 TANK HEATER RATING
         1=',F6.1)
C
```

```
C          *** COMPUTE FLUID DENSITIES AT FINAL TEMP.AND PRES.CONDITIONS
C
      TKOMXT = 0.0
      TKNMXT = 0.0
      DO 30 I= 1,KCYCLE
      TKOMXT = AMAX1(TKOMXT,O2TEMP(I))
      TKNMXT = AMAX1(TKNMXT,N2TEMP(I))
   30 CONTINUE
C
      CALL DENSON(TKOMXT,TKFPRS(1),1,RHOEND(1),ZEO)
      CALL DENSON(TKNMXT,TKFPRS(2),2,RHOEND(2),ZEN)
C
C          *** COMPUTE WEIGHT OF RESIDUAL FLUIDS IN TANKS
C
      WTRSID(1) = (RHOEND(1)/RHOBEG(1)*(1.0-(RHOEND(1)/RHOBEG(1))))
     1 * (O2CONS + O2RES)
      WTRSID(2) = (RHOEND(2)/RHOBEG(2)*(1.0-(RHOEND(2)/RHOBEG(2))))
     1 * (N2CONS + N2RES)
C
C          *** COMPUTE VOLUME OF THE FLUID TANKS
C
      VOLTK(1) = (O2TOTU + WTRSID(1))/(0.97*(RHOBEG(1) - RHOEND(1)))
      VOLTK(2) = (N2TOTU + WTRSID(2))/(0.97*(RHOBEG(2) - RHOEND(2)))
C
C          *** COMPUTE AREA OF SPHERICAL FLUID TANKS
C
      ARETK(1) = 4.84 * (VOLTK(1)**0.667)
      ARETK(2) = 4.84 * (VOLTK(2)**0.667)
C
C          *** COMPUTE HEAT LEAK INTO FLUID TANKS
C
      QLKOTK = 0.0
      QLKNTK = 0.0
      I =  0
      DO 40 I1 = 2,NDCYCL,2
      I = I + 1
      CALL TCOND(TENVR,O2TEMP(I),SNBAR(1),SITHIK(1,1),SITYPE(1,1),TOCND)
      QO2LK(I) = TOCND * ARETK(1) * DCYCLE(I1)
      QLKOTK   = QLKOTK + QO2LK(I)
C
      CALL TCOND(TENVR,N2TEMP(I),SNBAR(2),SITHIK(2,1),SITYPE(2,1),TNCND)
      QN2LK(I) = TNCND * ARETK(2) * DCYCLE(I1)
      QLKNTK   = QLKNTK + QN2LK(I)
C
   40 CONTINUE
C
C          *** COMPUTE QUANTITY OF FLUIDS VENTED DURING MISSION INTERVALS
C          *** AND TOTAL FLUIDS VENTED DURING THE MISSION
C
      WVO2 = 0.0
      WVN2 = 0.0
      PVNTO = SVPRES(1,1)
      PVNTN = SVPRES(2,1)
      DO 50 I = 1,KCYCLE
      CSBVO2  = CSUBV(O2TEMP(I),POPO2,1)
      CSBVN2  = CSUBV(N2TEMP(I),POPN2,18)
      QREQDO(I) = ((VOLTK(1)*CSBVO2)/48.3) * (PVNTO - POPO2) * 144.0
      QREQDN(I) = ((VOLTK(2)*CSBVN2)/54.9) * (PVNTN - POPN2) * 144.0
```

```
      IF(QREQDO(I).GT.QO2LK(I)) GO TO 45
      DELQ = QO2LK(I) - QREQDO(I)
      WTVNTO(I) = DELQ/(CSBVO2 * O2TEMP(I) * ((PVNTO/POPO2) - 1.0))
   45 WVO2 = WVO2 + WTVNTO(I)
      IF(QREQDN(I).GT.QN2LK(I)) GO TO 47
      DELQN = QN2LK(I) - QREQDN(I)
      WTVNTN(I) = DELQN/(CSBVN2 * N2TEMP(I) * ((PVNTN/POPN2) - 1.0))
   47 WVN2 = WVN2 + WTVNTN(I)
      IF(WVO2.LT.0.0) WVO2 = 0.0
      IF(WVN2.LT.0.0) WVN2 = 0.0
C
   90 CONTINUE
C
C
C         *** COMPUTE QUANTITY FLUIDS LOADED INTO TANKS
C
      TOTWTL(1) = O2TOTU + WTRSID(1) + WVO2
      TOTWTL(2) = N2TOTU + WTRSID(2) + WVN2
C
C         *** COMPUTE DIAMETER OF FLUID TANKS - ASSUMED SPHERICAL
C
      DITK(1) = ((1.9098 * (TOTWTL(1)/RHOBEG(1)))**0.33) * 12.0
      DITK(2) = ((1.9098 * (TOTWTL(2)/RHOBEG(2)))**0.33) * 12.0
C
C         *** COMPUTE FLUID TANK INSULATION WEIGHT
C
      IT1 = SITYPE(1,1)
      IT2 = SITYPE(2,1)
C
      TIWT(1,1) = NOP(1,1) * ARETK(1) * RHOI(IT1) * SITHIK(1,1)/12.0
      TIWT(2,1) = NOP(2,1) * ARETK(2) * RHOI(IT2) * SITHIK(2,1)/12.0
C
C         *** COMPUTE DIAMETER OF FLUID TANK VACUUM JACKETS
C
      DIVJ(1) = DITK + 1.60
      DIVJ(2) = DITK + 1.85
C
C         *** COMPUTE WEIGHT OF FLUID TANK PRESSURE VESSELS
C
      MATL1 = SMTYPE(1,1)
      CALL FINTAB (NTBID(9)+MATL1)
      FTUX1 = MIPE(1,TKOMXT)
C
      THKMT1 = (1.0 * POPO2 * 2.0 * (DITK(1)/2.0))/FTUX1
      IF(THKMT1.LT.MINTHK(MATL1)) THKMT1 = MINTHK(MATL1)
C
      MATL2 = SMTYPE(2,1)
      CALL FINTAB(NTBID(9)+MATL2)
      FTUX2 = MIPE(1,TKNMXT)
C
      THKMT2 = (1.0 * POPN2 * 2.0* (DITK(2)/2.0))/FTUX2
      IF(THKMT2.LT.MINTHK(MATL2)) THKMT2 = MINTHK(MATL2)
C
      WTPV(1) = 1.35 * ARETK(1) * RHOL(MATL1) * (THKMT1/12.0)
C
      WTPV(2) = 1.35 * ARETK(2) * RHOL(MATL2) * (THKMT2/12.0)
C
C         *** COMPUTE WEIGHT OF VACUUM JACKETS FOR FLUID TANKS
```

```
C          ASSUME SPHERICAL VACUUM JACKET(HARD SHELL) 6061-T6 ALUMINUM
C
      MATL3 = 3
      CALL FINTAB (NTBID(9),MATL3)
      FTUX3 = MIPE(1,TENVR)
C
      THKMT3 = (1.0 * (POPO2/2.0)* (DIVJ(1)/2.0))/FTUX3
      IF(THKMT3.LT.MINTHK(MATL3)) THKMT3 = MINTHK(MATL3)
C
      THKMT4 = (1.0 * (POPN2/2.0) * (DIVJ(2)/2.0))/FTUX3
      IF(THKMT4.LT.MINTHK(MATL3)) THKMT4 = MINTHK(MATL3)
C
      WTVJO = (PI * (DIVJ(1)**2)/144.0) * RHOL(MATL3) * (THKMT3/12.0)
C
      WTVJN = (PI * (DIVJ(2)**2)/144.0) * RHOL(MATL3) * (THKMT4/12.0)
C
C          *** COMPUTE TOTAL WEIGHT OF TANK
C
      WTTOT(1) = WTPV(1) + WTVJO + TIWT(1,1)
      WTTOT(2) = WTPV(2) + WTVJN + TIWT(2,1)
C
C     ******************************************************************
C
C          *** OUTPUT THE DATA COMPUTED TO THIS POINT ***
C
      JP = PAGE(0)
      WRITE (IOT,6116)
      WRITE (IOT,6001)
      WRITE (IOT,6009)
 6009 FORMAT(/T3,'DETERMINE FLUID DENSITIES AT FINAL DUTY CYCLE CONDITIO
     1NS AND DETERMINE RESIDUAL FLUID QUANTIES')
      WRITE (IOT,600)
      WRITE (IOT,647) TKOMXT, TKNMXT
  647 FORMAT(T3,'TKOMXT =',T13,F8.2,T25,'TKNMXT =',T37,F8.2)
      WRITE (IOT,648) RHOEND(1), RHOEND(2)
  648 FORMAT(T3,'RHOEND-O2 =',T15,F8.4,T27,'RHOEND-N2 =',T42,F8.4)
      WRITE (IOT,649) WTRSID(1), WTRSID(2)
  649 FORMAT(T3,'WTRSID-O2 =',T15,F8.3,T27,'WTRSID-N2 =',T42,F8.3)
      WRITE (IOT,6010)
 6010 FORMAT(/T3,'DETERMINE FLUID TANK VOLUME AND SURFACE AREAS')
      WRITE (IOT,600)
      WRITE (IOT,650) VOLTK(1), VOLTK(2)
  650 FORMAT(T3,'VOLTK-O2  =',T15,F8.3,T27,'VOLTK-N2  =',T42,F8.3)
      WRITE (IOT,651) ARETK(1), ARETK(2)
  651 FORMAT(T3,'AREA-O2TK =',T15,F8.3,T27,'AREA-N2TK =',T42,F8.3)
      WRITE (IOT,6011)
 6011 FORMAT(/T3,'DETERMINE HEAT LEAK INTO FLUID TANKS AND QUANTITY OF V
     1ENTED FLUIDS IF VENTING IS REQUIRED')
      WRITE (IOT,600)
      WRITE (IOT,652) TOCND, TNCND
  652 FORMAT(T3,'TOCND  =',T13,F8.4,T25,'TNCND  =',T37,F8.4)
      WRITE (IOT,653) (QO2LK(N),N=1,KCYCLE)
  653 FORMAT(T3,'QO2LK  =',T13,10(F9.6,3X))
      WRITE (IOT,654) (QN2LK(N),N=1,KCYCLE)
  654 FORMAT(T3,'QN2LK  =',T13,10(F9.6,3X))
      WRITE (IOT,655) QLKOTK, QLKNTK
  655 FORMAT(T3,'QLKOTK =',T13,F9.6,T25,'QLKNTK =',T37,F9.6)
      WRITE (IOT,656) (QREQDO(N),N=1,KCYCLE)
```

```
  656 FORMAT(T3,'QREQDO =',T13,10(F8.3,4X))
      WRITE (IOT,657) (QREQDN(N),N=1,KCYCLE)
  657 FORMAT(T3,'QREQDN =',T13,10(F8.3,4X))
      WRITE (IOT,658) (WTVNTO(N),N=1,KCYCLE)
  658 FORMAT(T3,'WTVNTO =',T13,10(F8.3,4X))
      WRITE (IOT,659) (WTVNTN(N),N=1,KCYCLE)
  659 FORMAT(T3,'WTVNTN =',T13,10(F8.3,4X))
      WRITE (IOT,660) WVO2,WVN2
  660 FORMAT(T3,'WVO2   =',T13,F8.3,T25,'WVN2   =',T37,F8.3)
      WRITE (IOT,6012)
 6012 FORMAT(/T3,'DETERMINE QUANTITY OF FLUIDS LOADED INTO TANKS')
      WRITE (IOT,600)
      WRITE (IOT,661) TOTWTL(1), TOTWTL(2)
  661 FORMAT(T3,'O2-LOADED =',T17,F8.3,T30,'N2-LOADED =',T44,F8.3)
      WRITE (IOT,6013)
 6013 FORMAT(/T3,'DETERMINE PRESSURE VESSEL, VACUUM JACKET, INSULATION,
     1AND TOTAL FLUID TANK WEIGHTS')
      WRITE (IOT,600)
      WRITE (IOT,662) DITK(1), DITK(2)
  662 FORMAT(T3,'DITK-O2=',T15,F8.2,T30,'DITK-N2=',T42,F8.2)
      WRITE (IOT,663) TIWT(1,1), TIWT(2,1)
  663 FORMAT(T3,'TIWT-O2=',T15,F8.3,T30,'TIWT-N2=',T42,F8.3)
      WRITE (IOT,664) DIVJ(1), DIVJ(2)
  664 FORMAT(T3,'DIVJ-O2=',T15,F8.2,T30,'DIVJ-N2=',T42,F8.2)
      WRITE (IOT,665) ROFTU(1), ROFTU(2)
  665 FORMAT(T3,'RHOFTU-O2=',T15,F8.7,T30,'RHOFTU-N2=',T42,F8.7)
      WRITE (IOT,666) WTPV(1), WTPV(2)
  666 FORMAT(T3,'WTPV-O2  =',T15,F8.2,T30,'WTPV-N2  =',T42,F8.2)
      WRITE (IOT,667) WTVJO, WTVJN
  667 FORMAT(T3,'WTVJ-O2  =',T15,F8.2,T30,'WTVJ-N2  =',T42,F8.2)
      WRITE (IOT,668) WTTOT(1),WTTOT(2)
  668 FORMAT(T3,'WTTOT-O2 =',T15,F8.2,T30,'WTTOT-N2 =',T42,F8.2)
C
C     *******************************************************************
C
C          *** COMPUTE WEIGHT OF O2 HEAT EXCHANGER
C
      JX = 0
      JX = JX +1
      IGAS = 1
      IFIN = 0                                 @ NO FINS ON ELECTRIC HEX
C
      WDOTX(JX,IGAS)  = WDTOMX
      UCODE(JX,IGAS)  = HXCODE(JX,IGAS)
      HEXCIT(JX,IGAS) = TEMPO2
      HEXCOT(JX,IGAS) = TLSNOM(1)
      HEXCIP(JX,IGAS) = POPO2
      HSQREQ(JX,IGAS) = QDTOMX
      ELCPOW(JX,IGAS) = HWTOMX
C
      CALL HEXELC(    1,HEXCIT(JX,IGAS),HEXCOT(JX,IGAS),HEXCIP(JX,IGAS),
     1 HTRFLX(1),LINDIA(1),WDOTX(JX,IGAS),O2RHO(IMAX),IFIN,
     2 WHXTOT(JX,IGAS),HXCDLP(JX,IGAS),UOA(JX,IGAS),DH(JX,IGAS),
     3 HLNGTH(JX,IGAS))
C
C
C          *** COMPUTE WEIGHT OF N2 HEAT EXCHANGER
C
```

```
      IGAS = 2
      IFIN = 0                          $ NO FINS ON ELECTRIC HEX
C
      WDOTX(JX,IGAS)  = WDTNMX
      UCODE(JX,IGAS)  = HXCODE(JX,IGAS)
      HEXCIT(JX,IGAS) = TEMPN2
      HEXCOT(JX,IGAS) = TLSNOM(2)
      HEXCIP(JX,IGAS) = POPN2
      HSQREQ(JX,IGAS) = QDTNMX
      ELCPOW(JX,IGAS) = HWTNMX
C
      CALL HEXELC(  18,HEXCIT(JX,IGAS),HEXCOT(JX,IGAS),HEXCIP(JX,IGAS),
     1 HTRFLX(1),LINDIA(2),WDOTX(JX,IGAS),O2RHO(JMAX),IFIN,
     2 WHXTOT(JX,IGAS),HXCDLP(JX,IGAS),UOA(JX,IGAS),DH(JX,IGAS),
     3 HLNGTH(JX,IGAS))
C
C          ***  OUTPUT THE HEAT EXCHANGER DATA ***
C
      CALL OTPHXE
C
C ***********************************************************************
C
C          ***  OUTPUT THE POWER SUMMARY DATA  ***
C
      CALL OPTPOW
C
C ***********************************************************************
C
C          *** COMPUTE THE TANK ENERGY HISTORY AND HEATER DUTY CYCLE ***
C
      TIMINC = 1.0                      $ PERIOD DEFINED BY GAS FLOW
      PTANK1 = POPO2
      PTANK2 = POPN2
C
      L1 = 0
      DO 100 K1 = 1,NDCYCL,2
      L1 = L1 + 1
      TIM(L1) = DCYCLE(K1)
  100 CONTINUE
C
      L2 = 0
      DO 150 J1 = 2,NDCYCL,2
      L2 = L2 + 1
      TNONOP(L2) = DCYCLE(J1)
  150 CONTINUE
C
      TIME = 0.0
      TKOW = 0.0
      TKNW = 0.0
      Q1CUMO = 0.0
      Q2CUMN = 0.0
C
      DO 110 K = 1,KCYCLE
C
      IF(TIM(K).EQ.0.0) GO TO 160
C
      NTP = TIM(K) * 6.0
      WDTO30 = WDTO2(K)/6.0
```

```
      WDTN30 = WDTN2(K)/6.0
      JP = PAGE(0)
C
      WRITE (IOT,700)
  700 FORMAT(/T42,'*** COMPUTE TANK ENERGY HISTORY AND HEATER DUTY CYCLE
     1 ***'/)
C
      WRITE (IOT,701) K, TIM(K)
  701 FORMAT(/T5,'SYSTEM DUTY CYCLE NUMBER ',I4,T45,'CYCLE IS ',F6.1,' H
     1OURS IN LENGTH')
C
      GO TO 170
C
  160 CONTINUE
      IF(TNONOP(K).NE.0.0) GO TO 161
      JP = PAGE(0)
      WRITE (IOT,700)
      WRITE (IOT,162)
  162 FORMAT(T5,' ***** THERE IS AN ERROR IN THE DUTY CYCLE DATA INPUT *
     1****' /T5,' ** THE CALCULATION WILL CONTINUE, OMITTING THE NON-OPE
     2RATING PERIOD **')
      GO TO 110
C
  161 CONTINUE
C
      JP = PAGE(0)
      WRITE (IOT,700)
      WRITE (IOT,701) K, TNONOP(K)
      WRITE (IOT,163) TNONOP(K)
  163 FORMAT(/T5,' *** THE SYSTEM IN A NON-DEMAND DUTY CYCLE PERIOD FOR
     1',F6.2,' HOURS.'/T5,' ** THE ANALYSIS FOR THIS PERIOD CONSIDERS ON
     2LY SPACECRAFT GAS LEAKAGE')
C
      NTP = TNONOP(K) * 6.0
      WDTO30 = (0.21 * GLKRAT/24.0)/6.0
      WDTN30 = (0.79 * GLKRAT/24.0)/6.0
C
  170 CONTINUE
C
      WRITE (IOT,702) K
  702 FORMAT(/T3,'TIME',T11,'GAS ',T20,'PER-CENT',T33,'DENSITY',T44,
     1 'FLUID',T53,'THETA',T63,'PHI',T71,'Q/REQD',T81,'Q-CUM',T91,'TANK'
     2,T100,'Q-HTR.',T110,'HEATER'/T11,'FLOW',T20,'WITHDRAWN',T44,'TEMP.
     3',T91,'PRES.',T100,'REQD.',T110,'TIME-ON'/T3,'(MIN)',T11,'(LBS)',
     4,T20,'(PERCENT)',T33,'(LB/CF)',T45,'(-R-)',T53,'(B/LB)',T62,
     5 '(P-CF/B)',T72,'(BTU)',T81,'(BTU)',T91,'(PSIA)', T100,'(BTU)',
     6 T110,'(MIN.)'/T11,'O2/N2',T22,'O2/N2',
     7 T34,'O2/N2',T44,'O2/N2',T53,'O2/N2',T62,'O2/N2',T72,'O2/N2',T81,
     8 'O2/N2',T91,'O2/N2',T100,'O2/N2',T110,'O2/N2',T120,'CYCLE -',I4/)
C
      LPRES = 0
      IK = 0
C
      DO 120 I = 1,NTP
C
      IF(LPRES.EQ.1) GO TO 112
      IF(RPRTIM(K).GT.0.0) GO TO 111
C
```

******** ECLSS ********

```
      TIME = TIME + 10.0
C
      TKOW = TKOW + WDTO30
      TKODP = TKOW
      TKNW = TKNW + WDTN30
      TKNDP = TKNW
      GO TO 113
C
  111 LPRES = LPRES + 1
      RPTIME = RPRTIM(K) * 60.0
  116 CONTINUE
      TIME = TIME + (RPTIME/10.0)
      IK = IK + 1
      TKOW = TKOW + ((WDTO2(K) * RPRTIM(K))/10.0)
      TKODP = TKOW
      TKNW = TKNW + ((WDTN2(K) * RPRTIM(K))/10.0)
      TKNDP = TKNW
      WDTO30 = (WDTO2(K) * RPRTIM(K))/10.0
      WDTN30 = (WDTN2(K) * RPRTIM(K))/10.0
      IF(IK.EQ.10) IK = 0
      GO TO 113
C
  112 TIME = TIME + 10.0
C
      TKOW = TKOW + WDOTON(K)/6.0
      TKODP = TKOW
      TKNW = TKNW + WDOTNN(K)/6.0
      TKNDP = TKNW
      WDTO30 = WDOTON(K)/6.0
      WDTN30 = WDOTNN(K)/6.0
      GO TO 113
C
C
C
  113 CONTINUE
C
      PCOXW = TKODP/O2TOTU
      PCN2W = TKNDP/N2TOTU
C
C
      C1 = 144.0/(RHOBEG(1) * (1544.2546/31.9988))
      C2 = 144.0/(RHOBEG(2) * (1544.2546/28.0134))
      C3 = 1.0 - ((C1 * TKFPRS(1))/(ZFO * TKFTEM(1)))
      C4 = 1.0 - ((C2 * TKFPRS(2))/(ZFN * TKFTEM(2)))
C
C
      ORHO = RHOBEG(1) * (1.0 - (PCOXW * C3))
      NRHO = RHOBEG(2) * (1.0 - (PCN2W * C4))
C
      CALL FINTAB (NTBID(8))
      XTAB(1) = PTANK1
      XTAB(2) = ORHO
      OXTEM = MIPE(2,XTAB)
C
      CALL FINTAB (NTBID(41))
      XTAB(1) = PTANK2
      XTAB(2) = NRHO
      N2TEM = MIPE(2,XTAB)
```

```
*********      ECLSS    ********

      C
      C
            CALL PHTHON(OXTEM,ORHO,1,PHI,THETA)
            DQDM1 = THETA
            DPDU1 = PHI
      C
      C
            CALL PHTHON(N2TEM,NRHO,18,PHI,THETA)
            DQDM2 = THETA
            DPDU2 = PHI
      C
      C
            QDTTK1 = WDTO30 * DQDM1
            QDTTK2 = WDTN30 * DQDM2
            Q1CUMO = Q1CUMO + QDTTK1
            Q2CUMN = Q2CUMN + QDTTK2
      C
            CALL BETAB(OXTEM,ORHO,1,BETAO)
            CALL BETAB(N2TEM,NRHO,18,BETAN)
      C
            CALL CSUBP(OXTEM,PTANK1,1,CPO)
            CALL CSUBP(N2TEM,PTANK2,18,CPN)
      C
      C
            DELP1 = TIMINC * ((DPDU1/VOLTK(1)) * ((-CPO/BETAO)*WDTO30))
            DELP2 = TIMINC * ((DPDU2/VOLTK(2)) * ((-CPN/BETAN)*WDTN30))
      C
            PTANK2 = PTANK2 + DELP2
            PTANK1 = PTANK1 + DELP1
      C
            IF(HTRFLX(2).LE.0.0) GO TO 122
      C
            QHTR1 = PI * HTRDIA(1) * HTRLNG(1) * HTRFLX(2)/60.0    B PER MIN.
            QHTR2 = PI * HTRDIA(2) * HTRLNG(2) * HTRFLX(2)/60.0
            GO TO 123
      C
        122 CONTINUE
      C
            QHTR1 = (PI * HTRDIA(1) * HTRLNG(1) * HTRRA1)/60.0    B PER MIN.
            QHTR2 = (PI * HTRDIA(2) * HTRLNG(2) * HTRRA2)/60.0    B PER MIN.
      C
        123 CONTINUE
      C
            IF(PTANK1.LE.PSET1) GO TO 130
            GO TO 135
        130 QELC1 = Q1CUMO
            HTRON1 = QELC1/QHTR1
        135 IF(PTANK2.LE.PSET2) GO TO 140
            GO TO 118
        140 QELC2 = Q2CUMN
            HTRON2 = QELC2/QHTR2
      C
        118 CONTINUE
      C
            IF(PAGE(2)) GO TO 227
            GO TO 228
        227 WRITE (IOT,702) K
            JP = PAGE(6)
```

```
  228 CONTINUE
C
      WRITE (IOT,703) TIME,WDTO30  ,PCOXW,ORHO,OXTEM,DQDM1,DPDU1,QDTTK1,
     1 Q1CUMO,PTANK1,QELC1,HTRON1,WDTN30  ,PCN2W,NRHO,N2TEM,DQDM2,
     2 DPDU2,QDTTK2,Q2CUMN,PTANK2,QELC2,HTRON2
  703 FORMAT(T1,F9.2,T11,F6.3,T21,F6.5,T32,F8.5,T43,F6.2,T52,F6.2,T61,
     1 F6.2,T71,F6.1,T80,F7.0,T90,F7.2,T99,F7.0,T109,F7.2/T11,F6.3,T21,
     2 F6.5,T32,F8.5,T43,F6.2,T52,F6.2,T61,F6.2,T71,F6.1,T80,F7.0,T90,
     3 F7.2,T99,F7.0,T109,F7.2)
C
      IF(PTANK1.LE.PSET1) HTRON1 = 0.0
      IF(PTANK1.LE.PSET1) QELC1 = 0.0
      IF(PTANK1.LE.PSET1) Q1CUMO = 0.0
      IF(PTANK1.LE.PSET1) PTANK1 = POPO2
      IF(PTANK2.LE.PSET2) HTRON2 = 0.0
      IF(PTANK2.LE.PSET2) QELC2 = 0.0
      IF(PTANK2.LE.PSET2) Q2CUMN = 0.0
      IF(PTANK2.LE.PSET2) PTANK2 = POPN2
C
      IF(IK.GT.0) GO TO 116
C
  120 CONTINUE
C
  110 CONTINUE
C
C  ******************************************************************
C
C
C  ******************************************************************
C
C         *** DO THE ECLSS CONFIGURATION ANALYSIS ***
C
      CALL LSSCMP
C
C  ******************************************************************
C
      WRITE (IOT,6099)
 6099 FORMAT(////T25,12('*******')/T25,12('*******')//T25,'***** THE SUP
     1ERCRITICAL EC-LSS CALCULATIONS HAVE BEEN COMPLETED  *****'//T25,
     2 12('*******')/T25,12('*******')/////)
C
      RETURN
C
      END
```

********* SUBRØUTINE ENGINE

```
C                  * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                  * ROUTINE NAME - ENGINE WEIGT AND TOTAL                 *
C                  *                IMPULSE PROPELLANT WEIGHT              *
C                  *                CALCULATION ROUTINE                    *
C                  * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2          *
C                  * PROGRAMMER   - R. BOLLINGER 1943 102 26933           *
C                  * DATE CODED   - 3/9/70                                *
C                  * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ENGINE
C
      LOGICAL DIAG
      LOGICAL PAGE,JP
C
      INCLUDE CCNTRL
      INCLUDE CDCYCL
      INCLUDE CENG
      INCLUDE CIOUNT
      INCLUDE TABLOK
C
C         ***** CALCULATE THE ENGINE WEIGHT
C
      IF (DIAG(0,6HENGINE)) WRITE (IOT,6000) NENG,GITEMP,THRUST,PSUBC,
     1                                       EXPRAT
C
      IF(SYSNUM.EQ.5) GO TO 5
C
      CALL FINTAB (NTBID(1))
      XTAB(1) = GITEMP
      XTAB(2) = PSUBC
      XTAB(3) = THRUST
      ENGWT = MIPE(3,XTAB) * NENG
C
C         ***** CALCULATE THE SPECIFIC IMPULSE
C
      CALL FINTAB (NTBID(2))
      XTAB(1) = GITEMP
      XTAB(2) = MIXRAT
      ISP = MIPE(2,XTAB)
C
      GO TO 15
C
    5 CONTINUE
C
C     ***** CALCULATE THE OMS ENGINE WEIGHT
C
      CALL FINTAB (NTBID(10))
      XTAB(1) = PSUBC
      XTAB(2) = THRUST
      ENGWT = MIPE(2,XTAB) * NENG
C
C     ***** CALCULATE OMS SPECIFIC IMPULSE
C
      CALL FINTAB (NTBID(11))
      XTAB(1) = PSUBC
      XTAB(2) = MIXRAT
      ISP = MIPE(2,XTAB)
```

```
C
   15 CONTINUE
C
C
C        ***** CALCULATE THE FLOW RATE
C        ***** CALCULATE THE TOTAL IMPULSE PROPELLANT WEIGHT
C
      KNT = 0
      TIPWT = 0.0
      WDOT = 0.0
C
      DO 10 I1 = 1,NDCYCL,2
      KNT = KNT + 1
      IF(DCYCLE(I1)) ,10,
      WDOT1 = IABS(NEOP(KNT))*THRUST/(PSI(KNT)*ISP)
      TIPWT = TIPWT + WDOT1*DCYCLE(I1)
      WDOTJ(KNT,2) = WDOT1/(MIXRAT + 1.0)
      WDOTJ(KNT,1) = WDOT1 - WDOTJ(KNT,2)
      IF(NEOP(KNT).LT.0) GO TO 10
      WDOT = AMAX1(WDOT,WDOT1)
   10 CONTINUE
      WDOTI(2) = WDOT/(MIXRAT + 1.0)
      WDOTI(1) = WDOT - WDOTI(2)
C
      IF (DIAG(1,6HENGINE)) WRITE (IOT,6000) KNT,ISP,WDOT,TIPWT,ENGWT,
     1                        WDOTI,((WDOTJ(I,J),I=1,KNT),J=1,2)
C
      JP = PAGE(0)
C
      WRITE (6,6100)
      WRITE (6,6301) ISP,ENGWT,WDOT1,WDOTI(1),WDOTI(2),TIPWT
      RETURN
 6000 FORMAT ('+'14X,I5,10X,6E15.6/(8E15.6))
 6100 FORMAT(//T38,'*** INITIATE PROGRAM AND CHARACTERIZE CONSUMER PARAM
     1ETERS ***'///)
 6301 FORMAT(T60,'* COMPUTED ENGINE PARAMETERS *'//T45,'ENGINE ISP',
     1 T80,E15.8/T45,'ENGINE WEIGHT - (LBS)',T80,E15.8/T45,'TOTAL ENGINE
     2 FLOW - (LB/SEC)',T80,E15.8/T45,'ONE ENGINE OXID.FLOW RATE-(LB/SEC
     3)',T80,E15.8/T45,'ONE ENGINE FUEL FLOW RATE-(LB/SEC)',T80,E15.8/T4
     45,'THRUST IMPULSE PROPELLANT WGT.',T80,E15.8)
C
      END
```

******** FUNCTIØN FINDR

```
      FUNCTION FINDR(N)
      DIMENSION G(18)
      DATA   G           /48.31,766.8,40.67,766.8,42.01,96.35,28.62,
     1 96.35,28.62,55.81,40.67,90.77,16.78,37.0,11.90,33.50,386.3,
     2 55.161/
      FINDR=G(N)
      RETURN
      END
```

******** FUNCTIØN FING1

```
      FUNCTION FING1(T,D)
      COMMON/CEOS/G(41)
      COMMON /SCRH/ X(40)
C
C.... ROUTINE TO CALCULATE INTEGRAL((R/D-1/D**2(DP/DT)) DD)
C
C.... WRITTEN 7/18/71 A MYERS
C
      D2=D*D
      D3=D2*D
      D4=D3*D
      D5=D4*D
      D6=D5*D
      D7=D6*D
      D8=D7*D
      D9=D8*D
      D10=D9*D
      TS= SQRT(T)
      T2=T*T
      T3=T2*T
      T4=T3*T
      T5=T4*T
      GM=G(41)
      F= EXP(GM*D2)
      G1=F/(2.00*GM)
      G2=(F*D2-2.00*G1)/(2.00*GM)
      G3=(F*D4-4.00*G2)/(2.00*GM)
      G4=(F*D6-6.00*G3)/(2.00*GM)
      G5=(F*D8-8.00*G4)/(2.00*GM)
      G6=(F*D10-10.00*G5)/(2.00*GM)
      X( 1)=-D
      X( 2)=-D/(2.00*TS)
      X( 3)=0.0
      X( 4)=+D/T2
      X( 5)=2.00*D/T3
      X( 6)=-D2/2.00
      X( 7)=0.0
      X( 8)=D2/(2.00*T2)
      X( 9)=D2/T3
      X(10)=-D3/3.00
      X(11)=0.0
      X(12)=D3/(3.00*T2)
      X(13)=0.0
      X(14)=D5/(5.00*T2)
      X(15)= 2.00*D5/(5.00*T3)
      X(16)=D6/(6.00*T2)
      X(17)=D7/(7.00*T2)
      X(18)=2.00*D7/(7.00*T3)
      X(19)=D8/(4.00*T3)
      X(20)=2.00*G1/T3
      X(21)=3.00*G1/T4
      X(22)=2.00*G2/T3
      X(23)=4.00*G2/T5
      X(24)=2.00*G3/T3
      X(25)=3.00*G3/T4
      X(26)=2.00*G4/T3
      X(27)=4.00*G4/T5
```

```
********    FINGI    ********

      X(28)=2.00*G5/T3
      X(29)=3.00*G5/T4
      X(30)=2.00*G6/T3
      X(31)=3.00*G6/T4
      X(32)=4.00*G6/T5
      FINGI=0.00
      DO 1 I=1,32
    1 FINGI=FINGI+G(I)*X(I)
      RETURN
      END
```

********* FUNCTIØN FING2

```
      FUNCTION FING2(T,D)
      COMMON/CEOS/G(41)
      COMMON /SCRH/ X(40)
C
C.... ROUTINE TO CALCULATE INTEGRAL((P/D**2-RT/D) DD)
C
      D2=D*D
      D3=D2*D
      D4=D3*D
      D5=D4*D
      D6=D5*D
      D7=D6*D
      D8=D7*D
      D9=D8*D
      D10=D9*D
      TS= SQRT(T)
      T2=T*T
      T3=T2*T
      T4=T3*T
      T5=T4*T
      GM=G(41)
      F= EXP(GM*D2)
      G1=F/(2.00*GM)
      G2=(F*D2-2.00*G1)/(2.00*GM)
      G3=(F*D4-4.00*G2)/(2.00*GM)
      G4=(F*D6-6.00*G3)/(2.00*GM)
      G5=(F*D8-8.00*G4)/(2.00*GM)
      G6=(F*D10-10.00*G5)/(2.00*GM)
      X( 1)=D*T
      X( 2)=D*TS
      X( 3)=D
      X( 4)=D/T
      X( 5)=D/T2
      X( 6)=D2*T/2.00
      X( 7)=D2/2.00
      X( 8)=D2/(2.00*T)
      X( 9)=D2/(2.00*T2)
      X(10)=D3*T/3.00
      X(11)=D3/3.00
      X(12)=D3/(3.00*T)
      X(13)=D4/4.00
      X(14)=D5/(5.00*T)
      X(15)=D5/(5.00*T2)
      X(16)=D6/(6.00*T)
      X(17)=D7/(7.00*T)
      X(18)=D7/(7.00*T2)
      X(19)=D8/(8.00*T2)
      X(20)=G1/T2
      X(21)=G1/T3
      X(22)=G2/T2
      X(23)=G2/T4
      X(24)=G3/T2
      X(25)=G3/T3
      X(26)=G4/T2
      X(27)=G4/T4
      X(28)=G5/T2
      X(29)=G5/T3
```

******** FING2 ********

```
      X(30)=G6/T2
      X(31)=G6/T3
      X(32)=G6/T4
      FING2=0.00
      DO 1 I=1,32
    1 FING2=FING2+G(I)*X(I)
      RETURN
      END
```

********* FUNCTIØN FING3

```
      FUNCTION FING3(T,D)
      COMMON/CEOS/G(41)
      COMMON /SCRH/ X(40)
C
C.... ROUTINE TO CALCULATE INTEGRAL((T/D**2)*(D2P/DT2) DD)
C
      D2=D*D
      D3=D2*D
      D4=D3*D
      D5=D4*D
      D6=D5*D
      D7=D6*D
      D8=D7*D
      D9=D8*D
      D10=D9*D
      TS= SQRT(T)
      T2=T*T
      T3=T2*T
      T4=T3*T
      T5=T4*T
      GM=G(41)
      F= EXP(GM*D2)
      G1=F/(2.00*GM)
      G2=(F*D2-2.00*G1)/(2.00*GM)
      G3=(F*D4-4.00*G2)/(2.00*GM)
      G4=(F*D6-6.00*G3)/(2.00*GM)
      G5=(F*D8-8.00*G4)/(2.00*GM)
      G6=(F*D10-10.00*G5)/(2.00*GM)
      X( 1)=0.0
      X( 2)=-D/(4.00*TS)
      X( 3)=0.0
      X( 4)=2.00*D/T2
      X( 5)=6.00*D/T3
      X( 6)=0.000
      X( 7)=0.0
      X( 8)=D2/T2
      X( 9)=3.00*D2/T3
      X(10)=0.000
      X(11)=0.0
      X(12)=(2.00*D3)/(3.00*T2)
      X(13)=0.0
      X(14)=(2.00*D5)/(5.00*T2)
      X(15)=(6.00*D5)/(5.00*T3)
      X(16)=D6/(3.00*T2)
      X(17)=(2.00*D7)/(7.00*T2)
      X(18)=(6.00*D7)/(7.00*T3)
      X(19)=(3.00*D8)/(4.00*T3)
      X(20)=6.000*G1/T3
      X(21)=12.00*G1/T4
      X(22)=6.000*G2/T3
      X(23)=20.00*G2/T5
      X(24)=6.000*G3/T3
      X(25)=12.00*G3/T4
      X(26)=6.000*G4/T3
      X(27)=20.00*G4/T5
      X(28)=6.000*G5/T3
      X(29)=12.00*G5/T4
```

******** FING3 ********

```
      X(30)=6.000*G6/T3
      X(31)=12.00*G6/T4
      X(32)=20.00*G6/T5
      FING3=0.00
      DO 1 I=1,32
    1 FING3=FING3+G(I)*X(I)
      RETURN
      END
```

********* SUBRØUTINE FINTAB

```
C                   * * * * * * * * * * * * ** * * * * * * * * * * *
C                   * ROUTINE NAME - FIND AND INPUT THE MASTER       *
C                   *                TABLE OF A SET OF TABLES        *
C                   * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2    *
C                   * PROGRAMMER   - R. BOLLINGER 1943 102 26933     *
C                   * DATE CODED   - 3/9/70                          *
C                   * * * * * * * * * * * * ** * * * * * * * * * * *
C
      SUBROUTINE FINTAB(MTN)
C
C          ***** EXPLANATION OF THE CALLING SEQUENCE
C              *
C              * MTN - MASTER TABLE NUMBER, THE INDEX OF THE ARRAY TLA
C              *         (SEE PDP CTAB) WHICH POINTS TO THE PROPER DRUM
C          *****         LOCATION FOR THE TABLE.
C
      LOGICAL DIAG
C
      INCLUDE CIOUNT
      INCLUDE CTAB
      INCLUDE CTABA
C
      IF (DIAG(0,6HFINTAB)) WRITE (IOT,6010) MTN,TLA(MTN)
      JTABID = MTN
      IF (TLA(MTN) .GT. 0) GO TO 5
      WRITE (IOT,6020) MTN
      CALL EXIT
    5 CONTINUE
      IDX1 = TLA(MTN)
      ND = ITABLE(IDX1)
      IF(ND.LE.2) GO TO 20
C
      NDM2 = ND - 2
      DO 10 I1=1,NDM2
      IDX1 = IDX1 + 1
      NP = ITABLE(IDX1)
      ITAB(1,I1) = NP
      DO 10 I2=1,NP
      IDX1 = IDX1 + 1
      TAB(I2+1,I1) = TABLE(IDX1)
   10 CONTINUE
C
   20 IDX1 = IDX1 + 1
      IF (DIAG(1,6HFINTAB)) WRITE (IOT,6010) ND,((TAB(I,J),J=1,5),I=1,6)
      RETURN
C
 6010 FORMAT ('+'14X,I7,5I15/(30X5E15.6))
C
 6020 FORMAT ('0'// 20X '***  LOAD TABLE NO.' I4,' BEFORE YOU TRY TO USE
     1 IT..  ***')
      END
```

B-147

LMSC-A991396

********* FUNCTIØN FLØDEQ

```
C                 * * * * * * * * * * * * * * * * * * * * * * * *
C                 * ROUTINE NAME - FANO LINE - ONE DIMENSIONAL  *
C                 *                COMPRESSIBLE-FLOW FUNCTION   *
C                 *                FOR ADIABATIC FLOW AT        *
C                 *                CONSTANT AREA WITH FRICTION  *
C                 * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                 * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                 * DATE CODED   - 3/10/70                      *
C                 * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION FLODEQ(M,G)
C
C         ***** EXPLANATION OF THE CALLING SEQUENCE
C             *
C             * M - MACH NUMBER
C             *
C             * G - SPECIFIC HEAT AT CONSTANT PRESSURE/
C         *****     SPECIFIC HEAT AT CONSTANT VOLUME
C
      REAL M,MSQ
C
      MSQ = M**2
      FLODEQ = (1.0 - MSQ)/(G*MSQ) + (G + 1.0)*ALOG((G + 1.0)*MSQ/
     1         (2.0 + (G - 1.0) * MSQ))/(2.0 * G)
      RETURN
      END
```

********* SUBRØUTINE FLØRAT

```
C                     * * * * * * * * * * ** * * * * * * * * * * * *
C                     * ROUTINE NAME - TOTAL FLOW RATE FOR H2 AND   *
C                     *                 O2 COMPUTATION.             *
C                     * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                     * PROGRAMMER   - J.A. MCKAY 19-43 201 X45178 *
C                     * DATE CODED   - FEBRUARY 1973     (REVISED) *
C                     * * * * * * * * * * ** * * * * * * * * * * * *
C
      SUBROUTINE FLORAT
C
      LOGICAL DIAG
      REAL ICIN,ICOUT
      REAL K1,K2,K3,K4,K5,K6,K7,K8
C
      INCLUDE CCNTRL
      INCLUDE CENG
      INCLUDE CFLRAT
      INCLUDE CHEX
      INCLUDE CPUMP
      INCLUDE CTANK
      INCLUDE CTURBN
      INCLUDE CIOUNT
C
      DIMENSION EQ13(2),EQ24(2),EQ57(2),EQ68(2)
C
      EQUIVALENCE (EQ13,K1),(EQ24,K2),(EQ13(2),K3),(EQ24(2),K4)
     1           ,(EQ57,K5),(EQ68,K6),(EQ57(2),K7),(EQ68(2),K8)
C
C
      IF(DIAG(0,6HFLORAT))WRITE(IOT,6020) WDOTI, TOTEMP, TITEMP, TEFF,
     1                                     EPDELP, TMRATO
      IF (DIAG(0,6HFLORAT)) WRITE (IOT,6020) HEXHOT(1,1),HEXHOT(1,2),
     1  HEXCOT(1,1), HEXCOT(1,2), HXMRAT(1,1), HXMRAT(1,2), HEXHIP(1,1)
     2,  HEXHIP(1,2)
C
C
      I3 = 1
      ISFLG = 1
      IF (SYSNUM .EQ. 2) I3 = 2
      IF (SYSNUM .EQ. 2 .AND. SCRIT .EQ. 2) ISFLG = 2
      DO 20 I1=1,2
      GO TO (14,12),ISFLG
   12 TERM1 = 0.
      TERM2 = 0.
      IF (HEXHIP(2,I1) .EQ. 0.) GO TO 18
   14 CONTINUE
C                     TERM FOR HEAT EXCHANGER NO. 1
      AVGTH = (HEXHIT(I3,I1) + HEXHOT(I3,I1)) / 2.0
      CALL CSUBP1 (AVGTH,HXMRAT(I3,I1),CPHEX)
      DIH = CPHEX * (HEXHIT(I3,I1) - HEXHOT(I3,I1))
      CALL ENTHOH (HEXCIP(I3,I1),HEXCIT(I3,I1),I1,ICIN)
      CALL ENTHOH (HEXCOP(I3,I1),HEXCOT(I3,I1),I1,ICOUT)
      DIC = ICOUT - ICIN
      TERM1 = DIC / (DIH*(1.0+HXMRAT(I3,I1)))
      GO TO (16,18),ISFLG
   16 CONTINUE
C                     TERM FOR PUMP TURBINE
```

```
*********     FLORAT     *********

      AVGTT = (TITEMP(II) + TOTEMP(II)) / 2.0
      CALL CSUBPI (AVGTT,TMRATO(II),CPEP)
      DELHTP = CPEP * (TITEMP(II) - TOTEMP(II))
      CALL RHOLIO (SITEMP(II,I),II,RHOLQ)
      TERM2 = 0.185*EPDELP(II) / (RHOLQ*PEFF(II)*TEFF(II)*DELHTP
     1                              *(1.0+TMRATO(II)))
   18 CONTINUE
C
C                   CALCULATE THE K-TERMS
      EQ13(II) = TERM1 * HXMRAT(I3,II)
      EQ24(II) = TERM1
      EQ57(II) = TERM2 * TMRATO(II)
      EQ68(II) = TERM2
   20 CONTINUE
C
      DENMHX = K1 + K4 + K2*K3 - K1*K4 - 1.0
      DENMTB = K5 + K8 + K6*K7 - K5*K8 - 1.0
C
      WFLO3 = (WDOTI   *K4 - WDOTI     - WDOTI(2)*K3) / DENMHX
      WFLO4 = (WDOTI(2)*K1 - WDOTI(2) - WDOTI    *K2) / DENMHX
C
      WFLO7 = (WDOTI   *K8 - WDOTI     - WDOTI(2)*K7) / DENMTB
      WFLO8 = (WDOTI(2)*K5 - WDOTI(2) - WDOTI    *K6) / DENMTB
C
      WFLO1 = K1 * WFLO3
      WFLO2 = K3 * WFLO4
      WFLO3 = K2 * WFLO3
      WFLO4 = K4 * WFLO4
C
      WFLO5 = K5 * WFLO7
      WFLO6 = K7 * WFLO8
      WFLO7 = K6 * WFLO7
      WFLO8 = K8 * WFLO8
C
C                   SET UP CONDITIONING FLOW RATE EQUATIONS
      WDHXTO = WFLO1 + WFLO2
      WDHXTF = WFLO3 + WFLO4
C
      WDTPTO = WFLO5 + WFLO6
      WDTPTF = WFLO7 + WFLO8
C
      WDOTT(1) = WDOTI(1) + WDHXTO + WDTPTO
      WDOTT(2) = WDOTI(2) + WDHXTF + WDTPTF
C
      GO TO (30,32),ISFLG
C                   OUTPUT TURBINE G.G. FLOWRATES
   30 CALL OTPFLT
      GO TO 34
C                   OUTPUT HEAT EXCHANGER G.G. FLOWRATES
   32 CALL OTPFLX
C
   34 CONTINUE
C
      IF (DIAG(1,6HFLORAT)) WRITE (IOT,6020) K1,K2,K3,K4,K5,K6,K7,K8,
     1                                        DENMHX,DENMTB
C
      RETURN
C
```

```
*********     FLORAT     *********
       6020 FORMAT ('+'14X,6E15.6/(19X6E15.6))
      C
            END
```

******** SUBRØUTINE FUELCL

```
              SUBROUTINE FUELCL
C
        LOGICAL  JP,PAGE
C
              INCLUDE CACCUM
              INCLUDE CAPU
              INCLUDE CDCYCL
              INCLUDE CENG
              INCLUDE CFUEL
              INCLUDE CHEX
              INCLUDE CHSORC
              INCLUDE CIOUNT
              INCLUDE CMATRL
              INCLUDE CONST
              INCLUDE CPUMP
              INCLUDE CTANK
              INCLUDE TABLOK
C
        DIMENSION TIM(12)
        DIMENSION WDOTX(MHX,2)
        EQUIVALENCE (WDOTCF,WDOTX)
C
C ***********************************************************************
C
        JP = PAGE(0)
C
        WRITE (IOT,6000)
  6000  FORMAT(//T38,'*** INITIATE PROGRAM AND CHARACTERIZE FUEL CELL PARA
       1METERS ***')
        WRITE (IOT,6001)
  6001  FORMAT(              //T2'PARAMETER',T14,'CYCLE-1',T26,'CYCLE-2',
       1 T38,'CYCLE-3',T50,'CYCLE-4',T62,'CYCLE-5',T74,'CYCLE-6',T86,
       2 'CYCLE-7',T98,'CYCLE-8',T110,'CYCLE-9',T121,'CYCLE-10'/)
C
C
C ***********************************************************************
C
C       BEGIN COMPUTATIONS FOR SUPERCRITICAL STORAGE
C
C ***********************************************************************
C
C             *** SET SELECTED INPUT PARAMETERS FOR CONVENIENCE
C
          PCOFC = SOPRES(1,1)
          PCHFC = SOPRES(2,1)
C
C             *** COMPUTE TOTAL ELECTRICAL POWER SUPPLIED FOR MISSION.
C
        POWTOT = 0.0
        I = 0
        DO 5 II = 1,NDCYCL,2
        I = I + 1
        POWTOT = POWTOT + PKW(I) * DCYCLE(II) * NEOP(I)   @ POWER IN KW-HRS
      5 CONTINUE
C
C             *** COMPUTE QUANTITY OF REACTANTS CONSUMMED FOR POWER
C             *** COMPUTE FLOWRATES OF EACH REACTANT FOR EACH INTERVAL
```

******** FUELCL ********

```
C
      WRFORP = 0.0
      I = 0
      DO 10 II = 1,NDCYCL,2
      I = I + 1
      WRP(I) = PKW(I) * DCYCLE(II) * SRCFC * NEOP(I)
      WRFORP = WRFORP + WRP(I)
C
   10 CONTINUE
C
C         *** OXYGEN CONSUMPTION
C
      C1 = MRFC/(MRFC + 1.0)
      WOCONS= WRFORP * C1
C
C         *** HYDROGEN CONSUMPTION
C
      C2 = 1.0/(MRFC + 1.0)
      WHCONS= WRFORP * C2
C
      DO 11 I = 1,KCYCLE
      WORFP(I) = WRP(I) * C1
      WHRFP(I) = WRP(I) * C2
C
      WDTFCO(I) = PKW(I) * SRCFC * C1
      WDTFCH(I) = PKW(I) * SRCFC * C2
   11 CONTINUE
C
C         *** MAX O2 AND H2 FLOW RATES
C
          PKWMAX = 0.0
          DO 15 I = 1,KCYCLE
          IF(PKW(I).LT.PKWMAX) GO TO 15
          PKWMAX = PKW(I)
   15     CONTINUE
C
          WDOTMX(1) = SRCFC * PKWMAX * C1                    @ LBS. PER HR.
          WDOTMX(2) = SRCFC * PKWMAX * C2                    @ LBS. PER HR.
C
      WDOTI(1) = WDOTMX(1)/3600.0
      WDOTI(2) = WDOTMX(2)/3600.0
C
      WRITE(IOT,6004)
 6004 FORMAT(T4,'COMPUTE TOTAL POWER AND FLOW RATES')
      WRITE (IOT,672)
  672 FORMAT(' ')
C
      WRITE (IOT,601) POWTOT, WRFORP
  601 FORMAT(T4,'POWTOT=',T13,F8.2,T25,'WRFORP=',T37,F8.1)
      WRITE (IOT,602) (WRP    (J),J=1,KCYCLE)
  602 FORMAT(T4,'WRP    =',T13,10(F8.3,4X))
      WRITE (IOT,603) C1, C2
  603 FORMAT(T4,'C1     =',T13,F8.3,T25,'C2     =',T37,F8.3)
      WRITE (IOT,604) WOCONS, WHCONS
  604 FORMAT(T4,'WOCONS=',T13,F8.2,T25,'WHCONS=',T37,F8.2)
      WRITE (IOT,674) (WORFP(J),J=1,KCYCLE)
  674 FORMAT(T4,'WORFP =',T13,10(F8.2,4X))
      WRITE (IOT,675) (WHRFP(J),J=1,KCYCLE)
```

```
  675 FORMAT(T4,'WHREP =',T13,10(F8.2,4X))
      WRITE (IOT,605) (WDTFCO(J),J=1,KCYCLE)
  605 FORMAT(T4,'WDTFCO=',T13,10(F8.2,4X))
      WRITE (IOT,606) (WDTFCH(J),J=1,KCYCLE)
  606 FORMAT(T4,'WDTFCH=',T13,10(F8.2,4X))
      WRITE (IOT,609) WDOTMX(1), WDOTMX(2)
  609 FORMAT(T4,'WDOTMAX-O2 =',F10.3,T30,'WDOTMAX-H2 =',F10.3)
      WRITE (IOT,673) WDOTI(1), WDOTI(2)
  673 FORMAT(T4,'WDOTI-O2 =',F12.8,T30,'WDOTI-H2 =',F12.8)
C
C  ***********************************************************************
C
C          *** DETERMINE AVERAGE SPECIFIC HEAT OF FREON COOLANT
C
      TMF21 = (TF21IN + TF21OU)/2.0
      CF21= CSPF21(TMF21)
      QF21 = CF21 * (TF21IN - TF21OU)          $ HEAT VALUE BTU/LB
C
C  ***********************************************************************
C
C          *** COMPUTE TOTAL HEAT REJECTED AND HEAT REJECTED FOR EACH
C          *** OPERATING INTERVAL BY FUEL CELLS
C
      QFCTOT = 0.0
      I = 0
      DO 40 I2 = 1,NDCYCL,2
      I = I + 1
      QAVAIL(I) = QDTFC * PKW(I) * DCYCLE(I2) * NEOP(I)
C
      WDTF21(I) = QAVAIL(I)/QF21
C
      QFCTOT = QFCTOT + QAVAIL(I)
   40 CONTINUE
C
      WRITE (IOT,672)
      WRITE(IOT,6005)
 6005 FORMAT(T4,'COMPUTE FUEL CELL HEAT REJECTION DATA')
      WRITE (IOT,672)
C
      WRITE (IOT,607) PKWMAX, TMF21
  607 FORMAT(T4,'PKWMAX=',T13,F8.2,T25,'TMF21 =',T37,F8.2)
      WRITE (IOT,608) QF21, CF21
  608 FORMAT(T4,'QF21  =',T13,F8.2,T25,'CF21  =',T37,F8.2)
      WRITE (IOT,610) (QAVAIL(J),J=1,KCYCLE)
  610 FORMAT(T4,'QAVAIL=',T13,10(F10.0,2X))
      WRITE (IOT,611) (WDTF21(J),J=1,KCYCLE)
  611 FORMAT(T4,'WDTF21=',T13,10(F10.1,2X))
      WRITE (IOT,612) QFCTOT
  612 FORMAT(T4,'QFCTOT=',T13,F10.1)
C
C  ***********************************************************************
C
C          *** DETERMINE ENTHALPY OF FLUIDS FEEDING FUEL CELL
C
           HFCO = OXENTH(PCOFC,TFCNOM(1))
           HFCH = HYENTH(PCHFC,TFCNOM(2))
C
C  ***********************************************************************
```

```
*********    FUELCL    *********

C
C        DETERMINE INITIAL TANK TEMPERATURES
C
         CALL FINTAB(NTBID(8))
         XTAB(1) = PCOFC
         XTAB(2) = 70.126
         TEMPO2 = MIPE(2,XTAB)
C
         CALL FINTAB(NTBID(7))
         XTAB(1) = PCHFC
         XTAB(2) = 4.365
         TEMPH2 = MIPE(2,XTAB)
C
C        DETERMINE INITIAL CSUBV VALUES FOR TANK T AND P CONDITIONS
C
         CISBVO = CSUBV(TEMPO2,PCOFC,1)
         CISBVH = CSUBV(TEMPH2,PCHFC,2)
C
C  COMPUTE THE COMPRESSIBILITY OF H2 AT TEMPERATURE TFH AND PRESSURE PFH
C
C  COMPUTE THE COMPRESSIBILITY OF O2 AT TEMPERATURE TFO AND PRESSURE PFO
C
         TKTO = 0.0
         TKTH = 0.0
         I = 0
         DO 450 II = 1,NDCYCL,2
         I = I + 1
C
         ZFO  = ZGET(TFOFC,PFOFC,1)
C
         CALL ZFIND(TFHFC,PFHFC,2,ZFH)
C
C  COMPUTE THE PERCENT OF USABLE H2 AND O2 WITHDRAWN UP TO THIS POINT
C  IN THE MISSION
C
         TKTO = TKTO + DCYCLE(II) * WDTFCO(I)
         TKO2WD(I) = TKTO
         TKTH = TKTH + DCYCLE(II) * WDTFCH(I)
         TKH2WD(I) = TKTH
C
C        ESTIMATE RESERVES AT 20 PERCENT
C
         PCWDO2(I) = TKO2WD(I)/(WOCONS + 0.2 * WOCONS)
         PCWDH2(I) = TKH2WD(I)/(WHCONS + 0.2 * WHCONS)
C
C  COMPUTE THE DENSITY OF H2 AND O2 AS A FUNCTION OF PERCENT WITHDRAWN.
C
         C3 = 1.0 - ((0.0427*PFHFC)/(ZFH*TFHFC))
         C4 = 1.0 - ((0.04253*PFOFC)/(ZFO*TFOFC))
C
         RHOTO2(I) = 70.126 * (1.0 - (PCWDO2(I) * C4))
         RHOTH2(I) = 4.365 * (1.0 - (PCWDH2(I) * C3))
C
C  ************************************************************************
C
C
C  COMPUTE THE TEMPERATURE OF O2 IN STORAGE TANK AS A FUNCTION OF
C  DENSITY AND PRESSURE.
```

```
*********    FUELCL    *********

      C
            CALL FINTAB (NTBID(8))
            XTAB(1) = PCOFC
            XTAB(2) = RHOTO2(I)
            TKO(I) = MIPE(2,XTAB)
      C
      C  COMPUTE THE TEMPERATURE OF H2 IN STORAGE TANK DURING TIME INTERVAL
      C  THETA(I) AS A FUNCTION OF DENSITY AND CONDITIONED PRESSURE.
      C
            CALL FINTAB (NTBID(7))
            XTAB(1) = PCHFC
            XTAB(2) = RHOTH2(I)
            TKH(I) = MIPE(2,XTAB)
      C
      C  ***********************************************************************
      C
      C          *** COMPUTE SPECIFIC HEAT INPUT (DQ/DM) FOR O2 AND H2 AS A
      C            * FUNCTION OF DENSITY AND STORED FLUID PRESSURE
      C
            CALL PHTHON(TKO(I),RHOTO2(I),1,PHI,THETA)
            DQDWO(I) = THETA
      C
            CALL FINTAB (NTBID(5))
            XTAB(1) = PCHFC
            XTAB(2) = RHOTH2(I)
            DQDWH(I) = MIPE(2,XTAB)
      C
      C  ***********************************************************************
      C
      C          *** SIZE O2 CONDITIONING HEAT EXCHANGER
      C
                 HTKO(I) = OXENTH(PCOFC,TKO(I))
                 Q1ODTR(I) = WDTFCO(I) * (HFCO - HTKO(I))
                WDT1FO(I) = Q1ODTR(I)/QF21
      C
      C          *** SIZE H2 CONDITIONING HEAT EXCHANGER
                 HTKH(I) = HYENTH(PCHFC,TKH(I))
                 Q1HDTR(I) = WDTFCH(I) * (HFCH - HTKH(I))
                WDT1FH(I) = Q1HDTR(I)/QF21
      C
      C  ***********************************************************************
      C
      C
      C          *** SIZE O2 TANK AND H2 TANK HEAT REQUIREMENTS
      C
            Q2ODTR(I) = WDTFCO(I) * DQDWO(I)
      C
            Q2HDTR(I) = WDTFCH(I) * DQDWH(I)
      C
            WDT2FO(I) = Q2ODTR(I)/QF21
      C
            WDT2FH(I) = Q2HDTR(I)/QF21
      C
      C  ***********************************************************************
      C
      C          *** COMPUTE THE ENERGY DIRIVATIVE (PHI) FOR O2 AND H2 AS A
      C            * FUNCTION OF PRESSURE AND TEMPERATURE FOR CONST VOL
      C
```

```
      CSBVFO(I) = CSUBV(TKO(I),PCOFC,1)
      CSBVFH(I) = CSUBV(TKH(I),PCHFC,2)
C
      CALL PHTHON(TKO(I),RHOTO2(I),1,PHI,THETA)
      PHIFO2(I) = PHI
C
      CALL FINTAB (NTBID(40))
      XTAB(1) = PCHFC
      XTAB(2) = RHOTH2(I)
      PHIFH2(I) = MIPE(2,XTAB)
C
  450 CONTINUE
      WRITE (IOT,672)
      WRITE(IOT,6006)
 6006 FORMAT(T4,'DETERMINE TANK CONDITIONS FOR DUTY CYCLE')
      WRITE (IOT,672)
C
      WRITE (IOT,613) HFCO, HFCH
  613 FORMAT(T4,'HFCO   =',T13,F8.2,T25,'HFCH   =',T37,F8.2)
      WRITE (IOT,614) TEMPO2, TEMPH2
  614 FORMAT(T4,'TEMPO2=',T13,F8.2,T25,'TEMPH2=',T37,F8.2)
      WRITE (IOT,615) CISBVO, CISBVH
  615 FORMAT(T4,'CISBVO=',T13,F8.3,T25,'CISBVH=',T37,F8.3)
      WRITE (IOT,616) (TKO2WD(J),J=1,KCYCLE)
  616 FORMAT(T4,'TKO2WD=',T13,10(F8.2,4X))
      WRITE (IOT,617) (TKH2WD(J),J=1,KCYCLE)
  617 FORMAT(T4,'TKH2WD=',T13,10(F8.2,4X))
      WRITE (IOT,618) (PCWDO2(J),J=1,KCYCLE)
  618 FORMAT(T4,'PCWDO2=',T13,10(F8.3,4X))
      WRITE (IOT,619) (PCWDH2(J),J=1,KCYCLE)
  619 FORMAT(T4,'PCWDH2=',T13,10(F8.3,4X))
      WRITE (IOT,620) C3, C4
  620 FORMAT(T4,'C3     =',T13,F8.2,T25,'C4     =',T37,F8.2)
      WRITE (IOT,621) (RHOTO2(J),J=1,KCYCLE)
  621 FORMAT(T4,'RHOTO2=',T13,10(F8.3,4X))
      WRITE (IOT,622) (RHOTH2(J),J=1,KCYCLE)
  622 FORMAT(T4,'RHOTH2=',T13,10(F8.3,4X))
      WRITE (IOT,623) (DQDWO(J),J=1,KCYCLE)
  623 FORMAT(T4,'DQDWO =',T13,10(F8.2,4X))
      WRITE (IOT,624) (DQDWH(J),J=1,KCYCLE)
  624 FORMAT(T4,'DQDWH =',T13,10(F8.2,4X))
      WRITE (IOT,625) (TKO(J),J=1,KCYCLE)
  625 FORMAT(T4,'TKO    =',T13,10(F8.2,4X))
      WRITE (IOT,626) (TKH(J),J=1,KCYCLE)
  626 FORMAT(T4,'TKH    =',T13,10(F8.2,4X))
C
      JP = PAGE(0)
      WRITE (IOT,6116)
 6116 FORMAT(//T42,'*** CONTINUE COMPUTATION OF FUEL CELL PARAMETERS ***
     1')
      WRITE (IOT,6001)
      WRITE(IOT,6007)
 6007 FORMAT(T4,'DETERMINE HEAT AND HOT FLOW REQUIREMENTS - EACH DUTY CY
     1CLE')
      WRITE (IOT,672)
C
      WRITE (IOT,627) (HTKO(J),J=1,KCYCLE)
  627 FORMAT(T4,'HTKO   =',T13,10(F8.3,4X))
```

```
      WRITE (IOT,628) (HTKH(J),J=1,KCYCLE)
  628 FORMAT(T4,'HTKH  =',T13,10(F8.3,4X))
      WRITE (IOT,629) (Q1ODTR(J),J=1,KCYCLE)
  629 FORMAT(T4,'Q1ODTR=',T13,10(F8.1,4X))
      WRITE (IOT,630) (Q1HDTR(J),J=1,KCYCLE)
  630 FORMAT(T4,'Q1HDTR=',T13,10(F8.1,4X))
      WRITE (IOT,631) (WDT1FO(J),J=1,KCYCLE)
  631 FORMAT(T4,'WDT1FO=',T13,10(F8.3,4X))
      WRITE (IOT,632) (WDT1FH(J),J=1,KCYCLE)
  632 FORMAT(T4,'WDT1FH=',T13,10(F8.3,4X))
      WRITE (IOT,633) (Q2ODTR(J),J=1,KCYCLE)
  633 FORMAT(T4,'Q2ODTR=',T13,10(F8.1,4X))
      WRITE (IOT,634) (Q2HDTR(J),J=1,KCYCLE)
  634 FORMAT(T4,'Q2HDTR=',T13,10(F8.1,4X))
      WRITE (IOT,635) (WDT2FO(J),J=1,KCYCLE)
  635 FORMAT(T4,'WDT2FO=',T13,10(F8.3,4X))
      WRITE (IOT,636) (WDT2FH(J),J=1,KCYCLE)
  636 FORMAT(T4,'WDT2FH=',T13,10(F8.3,4X))
C
      WRITE (IOT,672)
C
      WRITE(IOT,6008)
 6008 FORMAT(T4,'DETERMINE ENERGY DERIVATIVE -EACH CYCLE')
      WRITE (IOT,672)
C
      WRITE (IOT,637) (CSBVFO(J),J=1,KCYCLE)
  637 FORMAT(T4,'CSBVFO=',T13,10(F8.3,4X))
      WRITE (IOT,638) (CSBVFH(J),J=1,KCYCLE)
  638 FORMAT(T4,'CSBVFH=',T13,10(F8.3,4X))
      WRITE (IOT,639) (PHIFO2(J),J=1,KCYCLE)
  639 FORMAT(T4,'PHIFO2=',T13,10(F8.3,4X))
      WRITE (IOT,640) (PHIFH2(J),J=1,KCYCLE)
  640 FORMAT(T4,'PHIFH2=',T13,10(F8.3,4X))
C
C ***************************************************************************
C
C        *** CHECK TO ASSURE ADEQUATE SUPPLY OF FUEL CELL REJECT HEAT
C
         QTOTR = 0.0
         DO 500 I = 1,KCYCLE
      QSUMR(I) = Q1ODTR(I) + Q1HDTR(I) + Q2ODTR(I) + Q2HDTR(I)
         DQANET(I) = QAVAIL(I) - QSUMR(I)
      QTOTR = QTOTR + QSUMR(I)
  500 CONTINUE
C
      QEXCES = QFCTOT - QTOTR
C
C        *** COMPUTE MAX FLOW RATE OF F21 COOLANT REQUIRED.
C
      WF21MX = 0.0
      DO 501 I = 1,KCYCLE
      IF(WDT1FO(I).LT.WF21MX) GO TO 502
      WF21MX = WDT1FO(I)
  502 IF(WDT1FH(I).LT.WF21MX) GO TO 503
      WF21MX = WDT1FH(I)
  503 IF(WDT2FO(I).LT.WF21MX) GO TO 504
      WF21MX = WDT2FO(I)
  504 IF(WDT2FH(I).LT.WF21MX) GO TO 501
```

```
********    FUELCL    ********
          WF2IMX = WDT2FH(I)
  501 CONTINUE
C
          DQAMIN = DQANET(1)
          DO 505 I = 2,KCYCLE
          IF(DQANET(I).EQ.0.0) GO TO 505
          DQAMIN = AMIN1(DQAMIN,DQANET(I))
  505 CONTINUE
C
C
          IF(DQAMIN) 506,506,507
C
  506 CONTINUE
C
  507 CONTINUE
C
          WRITE (IOT,672)
          WRITE(IOT,6009)
 6009 FORMAT(T4,'CHECK ADEQUACY OF REJECTED HEAT SUPPLY')
          WRITE (IOT,672)
C
          WRITE (IOT,641) (QSUMR(J),J=1,KCYCLE)
  641 FORMAT(T4,'QSUMR =',T13,10(F8.1,4X))
          WRITE (IOT,642) (DQANET(J),J=1,KCYCLE)
  642 FORMAT(T4,'DQANET=',T13,10(F10.0,2X))
          WRITE (IOT,643)  QTOTR, QEXCES
  643 FORMAT(T4,'QTOTR =',T13,F8.1,T25,'QEXCES=',T37,F12.0)
          WRITE (IOT,644)  WF2IMX, DQAMIN
  644 FORMAT(T4,'WF2IMX=',T13,F8.2,T25,'DQAMIN=',T37,F12.0)
C
C  *********************************************************************
C
C          *** COMPUTE THE WEIGHT OF PROPELLANT TANK HEATER CIRCULATING
C            * COMPRESSOR.  FIRST COMPUTE THE MAXIMUM HEAT FLOW RATE
C          *** REQUIRED INTO THE TANK
C
          TKOMAX = 0.0
          TKHMAX = 0.0
          QMXTKO = 0.0
          QMXTKH = 0.0
          DO 510 I = 1,KCYCLE
          QMXTKO = AMAX1(QMXTKO,Q2ODTR(I))
          QMXTKH = AMAX1(QMXTKH,Q2HDTR(I))
          TKOMAX = AMAX1(TKOMAX,TKO(I))
          TKHMAX = AMAX1(TKHMAX,TKH(I))
  510 CONTINUE
C
C          *** COMPUTE THE SPECIFIC HEAT AND DENSITY OF H2 AND O2 AT
C            * THE FINAL TEMPERATURE AND PRESSURE
C
          PRFCMN(1) = PFOFC
          PRFCMN(2) = PFHFC
C
          CALL DENSON(TKOMAX,PRFCMN(1),1,RHOFIN(1),ZEE)
          CALL GSDNST(2,TKHMAX,PRFCMN(2),RHOFIN(2))
          CALL CSUBP(TKOMAX,PRFCMN(1),1,CPFO)
          CALL CSUBP(TKHMAX,PRFCMN(2),2,CPFH)
C
```

```
      WDTCFO = QMXTKO/(CPFO * (TF2IIN - 100. - TKOMAX))
C
      WDTCFH = QMXTKH/(CPFH * (TF2IIN - 125. - TKHMAX))
C
       WOCMP = (0.01455 * DELTCP * WDTCFO)/RHOFIN(1)
      WCIRCP(1) = WOCMP
       WHCMP = (0.01455 * DELTCP * WDTCFH)/RHOFIN(2)
      WCIRCP(2) = WHCMP
C
      WRITE (IOT,672)
      WRITE(IOT,6010)
 6010 FORMAT(T4,'COMPUTE WEIGHT OF CIRCULATING PUMPS FOR TANKS')
      WRITE (IOT,672)
C
      WRITE (IOT,645)  TKOMAX, TKHMAX
  645 FORMAT(T4,'TKOMAX=',T13,F8.2,T25,'TKHMAX=',T37,F8.2)
      WRITE (IOT,646)  QMXTKO, QMXTKH
  646 FORMAT(T4,'QMXTKO=',T13,F8.1,T25,'QMXTKH=',T37,F8.1)
      WRITE (IOT,647)  RHOFIN(1), RHOFIN(2)
  647 FORMAT(T4,'RHOFIN-O2=',T15,F8.3,T30,'RHOFIN-H2=',T42,F8.3)
      WRITE (IOT,648)  CPFO, CPFH
  648 FORMAT(T4,'CPFO  =',T13,F8.3,T25,'CPFH  =',T37,F8.3)
      WRITE (IOT,649)  WDTCFO, WDTCFH
  649 FORMAT(T4,'WDTCFO=',T13,F8.2,T25,'WDTCFH=',T37,F8.2)
      WRITE (IOT,650)  WOCMP, WHCMP
  650 FORMAT(T4,'WOCMP =',T13,F8.2,T25,'WHCMP =',T37,F8.2)
C
C ***********************************************************************
C
C          *** COMPUTE RESERVE REACTANT QUANTITY
C
          POWMAX = 0.0
          I = 0
          DO 20 II = 1,NDCYCL,2
          I = I + 1
          POWMAX = POWMAX + PKWMAX * DCYCLE(II) * NEOP(I)
   20     CONTINUE
C
C          * SET RESERVE AT 11.5 PERCENT OF MISSION EXTRAPOLATED MAX
C            POWER REACTANT REQUIREMENT *
C
          WRMAX = SRCFC * POWMAX
C
          WRRSRV = SRCFC * POWMAX * 0.115
C
C
      WORSRV = WRRSRV * C1                                     @ LBS.
      WHRSRV = WRRSRV * C2                                     @ LBS.
C
C          *** COMPUTE WEIGHT OF RESIDUAL REACTANTS.
C
C
      WTRES(1) = (RHOFIN(1)/RHOFIL(1))*(1.0/(1.0-(RHOFIN(1)/RHOFIL(1))))
     1 * (WOCONS + WORSRV + WOVENT)
C
      WTRES(2) = (RHOFIN(2)/RHOFIL(2))*(1.0/(1.0-(RHOFIN(2)/RHOFIL(2))))
     1 * (WHCONS + WHRSRV + WHVENT)
C
```

```
C
C        *** COMPUTE WEIGHT OF PURGE REACTANTS FOR FUEL CELLS.
C
      AMPHRS = (POWTOT * 1000.0)/FCVOLT
C
      PURGAS(1) = PRGRAT(1) * PRGTIM(1) * (AMPHRS/PRGINT(1))
C
      PURGAS(2) = PRGRAT(2) * PRGTIM(2) * (AMPHRS/PRGINT(2))
C
      NPRGE1 = AMPHRS/PRGINT(1)
      IF(PRGINT(1).EQ.0.0) NPRGE1 = 0
      NPRGE2 = AMPHRS/PRGINT(2)
      IF(PRGINT(2).EQ.0.0) NPRGE2 = 0
C
      JP = PAGE(0)
      WRITE (IOT,6116)
      WRITE (IOT,6001)
      WRITE(IOT,6012)
 6012 FORMAT(T4,'COMPUTE RESERVE, RESIDUAL AND PURGE REACTANT QUANTITIES
     1')
      WRITE (IOT,672)
      WRITE (IOT,651)  POWMAX, WRMAX
  651 FORMAT(T4,'POWMAX=',T13,F8.1,T25,'WRMAX =',T37,F8.2)
      WRITE (IOT,652) WRRSRV, WORSRV, WHRSRV
  652 FORMAT(T4,'WRRSRV=',T13,F8.2,T25,'WORSRV=',T37,F8.2,T48,'WHRSRV=',
     1 T60,F8.2)
      WRITE (IOT,654)  WTRES(1), WTRES(2)
  654 FORMAT(T4,'WTRES-O2 =',F7.2,T25,'WTRES-H2 =',F7.2)
      WRITE (IOT,682) AMPHRS, NPRGE1, NPRGE2
  682 FORMAT(T4,'TOTAL AMPERE-HOURS =',F10.2,T40,'NUMBER OF FUEL CELL PU
     1RGES - FOR OXYGEN SIDE IS',I4,'-FOR HYDROGEN SIDE IS',I4)
      WRITE (IOT,683) PURGAS(1), PURGAS(2)
  683 FORMAT(T4,'QUANTITY OF PURGE GAS USED IS  -  OXYGEN',F6.2,'  - HYD
     1ROGEN',F6.2)
C
C        *** COMPUTE VOLUME OF THE REACTANT TANKS.
C
      VOLTNK(1) = (WOCONS + WORSRV + WOVENT + WTRES(1) + PURGAS(1))/(.97
     1 * (RHOFIL(1) - RHOFIN(1)))
C
      VOLTNK(2) = (WHCONS + WHRSRV + WHVENT + WTRES(2) + PURGAS(2))/(.98
     1 * (RHOFIL(2) - RHOFIN(2)))
C
C        *** COMPUTE AREA OF SPHERICAL REACTANT TANKS.
C
      AREATK(1) = 4.84 * (VOLTNK(1)**0.666)
C
      AREATK(2) = 4.84 * (VOLTNK(2)**0.666)
C
      WRITE (IOT,672)
      WRITE(IOT,6011)
 6011 FORMAT(T4,'COMPUTE TANK VOLUME AND SURFACE AREA')
      WRITE (IOT,672)
C
      WRITE (IOT,655)  VOLTNK(1), VOLTNK(2)
  655 FORMAT(T4,'VOLTNK-O2=',T15,F8.2,T30,'VOLTNK-H2=',T42,F8.2)
      WRITE (IOT,656)  AREATK(1), AREATK(2)
  656 FORMAT(T4,'AREATK-O2=',T15,F8.2,T30,'AREATK-H2=',T42,F8.2)
```

```
C
C          *** COMPUTE HEAT LEAK INTO REACTANT TANKS,-PER HOUR AND TOTAL
C
           QLEAKO = 0.0
           QLEAKH = 0.0
           I = 0
           DO 30 II = 2,NDCYCL,2
           I = I + 1
           CALL TCOND(TENV,TKO(I),SNBAR(1),SITHIK(1,1),SITYPE(1,1),TOCOND)
           QLKO(I) = TOCOND * AREATK(1) * DCYCLE(II)
           QLEAKO  = QLEAKO + QLKO(I)
C
           CALL TCOND(TENV,TKH(I),SNBAR(2),SITHIK(2,1),SITYPE(2,1),THCOND)
           QLKH(I) = THCOND * AREATK(2) * DCYCLE(II)
           QLEAKH  = QLEAKH + QLKH(I)
C
        30 CONTINUE
C
C   ***************************************************************************
C
C
C   BEGIN CALCULATING WEIGHT OF H2 VENTED DURING THE MISSION. FIRST
C   COMPUTE THE SPECIFIC HEAT OF H2 AT T= TKH(I) AND P= PCHFC
C
           WVHO = 0.0
           WVHH = 0.0
           PVENTO = SVPRES(1,1)
           PVENTH = SVPRES(2,1)
C
           DO 550 I = 1,KCYCLE
C
           CSBVO2 = CSUBV(TKO(I),PCOFC,1)
           CSBVH2 = CSUBV(TKH(I),PCHFC,2)
           QRQDO2 = ((VOLTNK(1)*CSBVO2)/48.3) * (PVENTO - PCOFC) * 144.0
           QRQDH2 = ((VOLTNK(2)*CSBVH2)/776.5) * (PVENTH - PCHFC) * 144.0
           IF(QRQDO2.GT.QLKO(I)) GO TO 31
           DELQ   = QLKO(I) - QRQDO2
           WVENTO = DELQ/(CSBVO2 * TKO(I) * ((PVENTO/PCOFC) - 1.0))
        31 WVHO   = WVHO + WVENTO
           IF(QRQDH2.GT.QLKH(I)) GO TO 32
           DELQH  = QLKH(I) - QRQDH2
           WVENTH = DELQH/(CSBVH2 * TKH(I) * ((PVENTH/PCHFC) - 1.0))
        32 WVHH   = WVHH + WVENTH
C
       550 CONTINUE
C
C   COMPUTE RESULTANT TOTAL WEIGHT OF VENTED GASES.
C
           IF(WVHO.LT.0.0) WVHO = 0.0
           IF(WVHH.LT.0.0) WVHH = 0.0
C
           WOVENT = WVHO
           WHVENT = WVHH
C
           WRITE (IOT,672)
           WRITE(IOT,6013)
      6013 FORMAT(T4,'COMPUTE HEAT LEAK AND VENTED REACTANT QUANTITY')
           WRITE (IOT,672)
```

```
C
      WRITE (IOT,657) (QLKO(J),J=1,KCYCLE)
  657 FORMAT(T4,'QLKO   =',T13,10(E9.4,3X))
      WRITE (IOT,658) (QLKH(J),J=1,KCYCLE)
  658 FORMAT(T4,'OLKH   =',T13,10(E9.4,3X))
      WRITE (IOT,659)  QLEAKO , QLEAKH
  659 FORMAT(T4,'QLEAKO=',T13,E10.5,T30,'QLEAKH=',T39,E10.5)
      WRITE (IOT,660)  WVHO, WVHH
  660 FORMAT(T4,'WVHO   =',T13,F8.2,T25,'WVHH   =',T37,F8.2)
      WRITE (IOT,661) WOVENT, WHVENT
  661 FORMAT(T4,'WOVENT=',T13,F8.2,T25,'WHVENT=',T37,F8.2)
C
C ***********************************************************************
C
C          *** COMPUTE REACTANT LOADED INTO TANKS
C
      WRTOTL(1) = (WOCONS + WORSRV + WOVENT + WTRES(1) + PURGAS(1))
C
      WRTOTL(2) = (WHCONS + WHRSRV + WHVENT + WTRES(2) + PURGAS(2))
C
C          *** COMPUTE DIAMETER OF REACTANT TANKS - ASSUMED SPHERICAL
C
      DIATK(1) = ((1.9098 * (WRTOTL(1)/RHOFIL(1)))**0.33) * 12.0
      DIATK(2) = ((1.9098 * (WRTOTL(2)/RHOFIL(2)))**0.33) * 12.0
C
      ITANK1 = 0
      IF(DIATK(1).GT.TKMXDI(1)) ITANK1 = 1
      ITANK2 = 0
      IF(DIATK(2).GT.TKMXDI(2)) ITANK2 = 1
C
C          *** COMPUTE REACTANT TANK INSULATION WEIGHT
C
      IT1 = SITYPE(1,1)
      IT2 = SITYPE(2,1)
C
      TIWT(1,1) = NOP(1,1) * AREATK(1) * RHOI(IT1) * SITHIK(1,1)/12.0
      TIWT(2,1) = NOP(2,1) * AREATK(2) * RHOI(IT2) * SITHIK(2,1)/12.0
C
C          *** COMPUTE DIAMETER OF REACTANT TANK VACUUM JACKETS
C
      DIAVJ(1) = DIATK(1) + (VJANUL(1) * 2.0)
      DIAVJ(2) = DIATK(2) + (VJANUL(2) * 2.0)
C
C          *** COMPUTE WEIGHT OF REACTANT TANK PRESSURE VESSELS
C
      MATL1 = SMTYPE(1,1)
      CALL FINTAB(NTBID(9)+MATL1)
      FTUX1 = MIPE(1,TKOMAX)
C
      THKMT1 = (1.0 * PCOFC * 2.0 * (DIATK(1)/2.0))/FTUX1
      IF(THKMT1.LT.MINTHK(MATL1)) THKMT1 = MINTHK(MATL1)
C
      MATL2 = SMTYPE(2,1)
      CALL FINTAB(NTBID(9)+MATL2)
      FTUX2 = MIPE(1,TKHMAX)
C
      THKMT2 = (1.0 * PCHFC * 2.0 * (DIATK(2)/2.0))/FTUX2
      IF(THKMT2.LT.MINTHK(MATL2)) THKMT2 = MINTHK(MATL2)
```

```
C
      WTPVT(1) = 1.35 * AREATK(1) * RHOL(MATL1) * (THKMT1/12.0)
C
      WTPVT(2) = 1.35 * AREATK(2) * RHOL(MATL2) * (THKMT2/12.0)
C
C         *** COMPUTE WEIGHT OF VACUUM JACKET FOR REACTANT TANKS
C         ASSUME SPHERICAL VACUUM JACKET (HARD SHELL) 6061-T6 ALUMINUM
C
      MATL3 = 3
      CALL FINTAB (NTBID(9),MATL3)
      FTUX3 = MIPE(1,TENV)
C
      THKMT3 = (1.0 * (PCOFC/2.0) * (DIAVJ(1)/2.0))/FTUX3
      IF(THKMT3.LT.MINTHK(MATL3)) THKMT3 = MINTHK(MATL3)
C
      THKMT4 = (1.0 * (PCHFC/2.0) * (DIAVJ(2)/2.0))/FTUX3
      IF(THKMT4.LT.MINTHK(MATL3)) THKMT4 = MINTHK(MATL3)
C
      WTVJ(1) = (PI * (DIAVJ(1)**2)/144.0) * RHOL(MATL3)*(THKMT3/12.0)
C
      WTVJ(2) = (PI * (DIAVJ(2)**2)/144.0) * RHOL(MATL3)*(THKMT4/12.0)
C
C         *** COMPUTE TOTAL WEIGHT OF TANK
C
      WTTOT(1) = WTPVT(1) + WTVJ(1) + TIWT(1,1)
      WTTOT(2) = WTPVT(2) + WTVJ(2) + TIWT(2,1)
C
      WRITE (IOT,672)
      WRITE(IOT,6014)
 6014 FORMAT(T4,'COMPUTE PRESSURE VESSEL, VACUUM JACKET, INSULATION AND
     1TOTAL TANK WEIGHTS')
      WRITE (IOT,672)
C
      WRITE (IOT,662)  WRTOTL(1), WRTOTL(2)
  662 FORMAT(T4,'WRTOTL-O2=',T15,F8.2,T30,'WRTOTL-H2=',T42,F8.2)
      WRITE (IOT,663)  DIATK(1), DIATK(2)
  663 FORMAT(T4,'DIATK-O2 =',T15,F8.2,T30,'DIATK-H2 =',T42,F8.2)
C
      IF(ITANK1.EQ.1) WRITE (IOT,680) DIATK(1),TKMXDI(1)
  680 FORMAT( '  *** O2 STORAGE TANK DIAMETER ',F6.2,' INCHES EXCEEDS TH
     1E MAXIMUM ALLOWABLE PRESSURE VESSEL DIAMETER OF ',F6.2,'INCHES,',/
     2 T5,'ADDITIONAL TANKAGE MUST BE PROVIDED. ***')
C
      IF(ITANK2.EQ.1) WRITE (IOT,681) DIATK(2),TKMXDI(2)
  681 FORMAT( '  *** H2 STORAGE TANK DIAMETER ',F6.2,' INCHES EXCEEDS TH
     1E MAXIMUM ALLOWABLE PRESSURE VESSEL DIAMETER OF ',F6.2,'INCHES,',/
     2 T5,'ADDITIONAL TANKAGE MUST BE PROVIDED. ***')
C
      WRITE (IOT,664)  TIWT(1,1), TIWT(2,1)
  664 FORMAT(T4,'TIWT-O2  =',T15,F8.2,T30,'TIWT-H2  =',T42,F8.2)
      WRITE (IOT,665)  DIAVJ(1), DIAVJ(2)
  665 FORMAT(T4,'DIAVJ-O2 =',T15,F8.2,T30,'DIAVJ-H2 =',T42,F8.2)
      WRITE (IOT,666)  RHOFTU(1), RHOFTU(2)
  666 FORMAT(T4,'RHOFTU-O2=',T15,F8.7,T30,'RHOFTU-H2=',T42,F8.7)
      WRITE (IOT,667)  WTPVT(1), WTPVT(2)
  667 FORMAT(T4,'WTPVT-O2 =',T15,F8.2,T30,'WTPVT-H2 =',T42,F8.2)
      WRITE (IOT,668)  WTVJ(1), WTVJ(2)
  668 FORMAT(T4,'WTVJ-O2  =',T15,F8.2,T30,'WTVJ-H2  =',T42,F8.2)
```

```
      WRITE (IOT,669)  WTTOT(1), WTTOT(2)
  669 FORMAT(T4,'WTTOT=O2 =',T15,F8.2,T30,'WTTOT=H2 =',T42,F8.2)
C
C   **************************************************************************
C
C          *** COMPUTE WEIGHT OF FUEL CELLS
C
      FCWGT = PKWMAX * SPWTFC * (NFCOP + NFCSTB)
      ENGWT = FCWGT
C
      WRITE (IOT,672)
      WRITE(IOT,6015) NFCOP, NFCSTB
 6015 FORMAT(T4,'COMPUTE WEIGHT OF FUEL CELLS',T35,I3,2X,'OPERATING AND'
     1,I3,2X,'STANDBY')
      WRITE (IOT,672)
C
      WRITE (IOT,670)  FCWGT
  670 FORMAT(T4,'FCWGT  =',T15,F8.2)
C
C   **************************************************************************
C
C          *** COMPUTE  WEIGHT OF HEAT EXCHANGERS
C            * FIRST COMPUTE THE HOT FLUID TEMPERATURE DROP
C          *** IN THE SERIES HEAT EXCHANGER SEQUENCE
C
           DTO1MX = 0.0
           DTH1MX = 0.0
           DTO2MX = 0.0
           DTH2MX = 0.0
C
           DO 100 I = 1,KCYCLE
           DTO1MX = AMAX1(DTO1MX,WDT1FO(I))
           DTH1MX = AMAX1(DTH1MX,WDT1FH(I))
           DTO2MX = AMAX1(DTO2MX,WDT2FO(I))
           DTH2MX = AMAX1(DTH2MX,WDT2FH(I))
  100      CONTINUE
      IF(DTO1MX.LT.DTO2MX) DTO1MX = DTO2MX
      IF(DTH1MX.LT.DTH2MX) DTH1MX = DTH2MX
      IF(DTO1MX.LT.DTH1MX) DTO1MX = DTH1MX
      WDTFMX = DTO1MX
C
C
      Q1ODMX = 0.0
      DO 110 I =1,KCYCLE
      IF(Q1ODTR(I).LT.Q1ODMX) GO TO 110
      Q1ODMX = Q1ODTR(I)
      IMAXO = I
  110 CONTINUE
C
      Q1HDMX = 0.0
      DO 120 I = 1,KCYCLE
      IF(Q1HDTR(I).LT.Q1HDMX) GO TO 120
      Q1HDMX = Q1HDTR(I)
      IMAXH = I
  120 CONTINUE
C
      T2F21 = TF21IN - (Q1ODTR(IMAXO)/(CPF21 * WDTFMX))
C
```

```
*********    FUELCL    *********

             T3F21 = T2F21  - (QMXTKO/(CP21 * WDTFMX))
             T4F21 = T3F21  - (Q1HDTR(IMAXH)/(CF21 * WDTFMX))
      C
             T5F21 = T4F21 - (QMXTKH/(CF21 * WDTFMX))
      C
             JX = 0
             QTREQ(1) = 0,0
             QTREQ(2) = 0.0
      C
      C          *** COMPUTE WEIGHT AND CHARACTERISTICS OF  O2 HEX BETWEEN
      C            *          O2 TANK AND FUEL CELL
      C
             JX = JX +1
             IGAS =1
      C
             WDOTX(JX,IGAS) = WDTIFO(IMAXO)              @  LBS PER HOUR
             UCODE(JX,IGAS) = HXCODE(JX,IGAS)
             HEXHIT(JX,IGAS) = TF21IN
             HEXCIT(JX,IGAS) = TKO(IMAXO)
             HEXHOT(JX,IGAS) = T2F21
             HEXCOT(JX,IGAS) = TFCNOM(1)
             HEXCOP(JX,IGAS) = PCOFC - 3.0
             HEXCIP(JX,IGAS) = PCOFC
      C
             HSQREQ(JX,IGAS) = Q1ODMX
             QTREQ(IGAS) = QTREQ(IGAS) + HSQREQ(JX,IGAS)
             CALL CSUBP(TKO(IMAXO),PCOFC,1,CPCLDF(JX,IGAS))
             TMF211 = (TF21IN + T2F21)/2.0
             CPF211 = CSPF21(TMF211)
             CPHOTF(JX,IGAS) = CPF211
             HSGCPE(JX,IGAS) = CPF211
             HSGTOT(JX,IGAS) = Q1ODMX/(CPF211 * (TF21IN - T2F21))
      C
             WDOTH(JX,IGAS) = Q1ODMX/(CPF211 * (TF21IN - T2F21))
             CALL HEXF21(IGAS,Q1ODMX,HEXCIP(JX,IGAS),WHXTOT(JX,IGAS))
      C
      C
      C
      C          *** COMPUTE WEIGHT AND CHARACTERISTICS OF H2 HEX BETWEEN
      C            *          H2 TANK AND FUEL CELL
      C
             IGAS = 2
      C
             WDOTX(JX,IGAS) = WDTIFH(IMAXH)              @  LBS PER HOUR
             UCODE(JX,IGAS) = HXCODE(JX,IGAS)
             HEXHIT(JX,IGAS) = T3F21
             HEXCIT(JX,IGAS) = TKH(IMAXH)
             HEXHOT(JX,IGAS) = T4F21
             HEXCOT(JX,IGAS) = TFCNOM(2)
             HEXCOP(JX,IGAS) = PCHFC - 3.0
             HEXCIP(JX,IGAS) = PCHFC
      C
             HSQREQ(JX,IGAS) = Q1HDMX
             QTREQ(IGAS) = QTREQ(IGAS) + HSQREQ(JX,IGAS)
             CALL CSUBP(TKH(IMAXH),PCHFC,2,CPCLDF(JX,IGAS))
             TMF212 = (T3F21 + T4F21)/2.0
             CPF212 = CSPF21(TMF212)
             CPHOTF(JX,IGAS) = CPF212
```

```
      HSGCPE(JX,IGAS) = CPF212
      HSGTOT(JX,IGAS) = QIHDMX/(CPF212 * (T3F21 - T4F21))
      WDOTH(JX,IGAS) = QIHDMX/(CPF212 * (T3F21 - T4F21))
C
      CALL HEXF21(IGAS,QIHDMX,HEXCIP(JX,IGAS),WHXTOT(JX,IGAS))
C
C
C         *** COMPUTE WEIGHT AND CHARACTERISTICS OF O2 HEX BETWEEN
C           *          O2 TANK CIRCULATOR AND O2 TANK
C
      JX= JX + 1
      IGAS = 1
C
      WDOTX(JX,IGAS) = WDTCFO
      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
      HEXHIT(JX,IGAS) = T2F21
      HEXCIT(JX,IGAS) = TKO(1)
      HEXHOT(JX,IGAS) = T3F21
      HEXCOT(JX,IGAS) = TKOMAX
      HEXCOP(JX,IGAS) = PCOFC
      HEXCIP(JX,IGAS) = PCOFC + DELTCP
C
      HSQREQ(JX,IGAS) = QMXTKO
      QTREQ(IGAS) = QTREQ(IGAS) + HSQREQ(JX,IGAS)
      CALL CSUBP(TKO(1),PCOFC,1,CPCLDF(JX,IGAS))
      TMF213 = (T2F21 + T3F21)/2.0
      CPF213 = CSPF21(TMF213)
      CPHOTF(JX,IGAS) = CPF213
      HSGCPE(JX,IGAS) = CPF213
      HSGTOT(JX,IGAS) = QMXTKO/(CPF213 * (T2F21 - T3F21))
      WDOTH(JX,IGAS) = QMXTKO/(CPF213 * (T2F21 - T3F21))
C
      CALL HEXF21(IGAS,QMXTKO,HEXCIP(JX,IGAS),WHXTOT(JX,IGAS))
C
C
C         *** COMPUTE WEIGHT AND CHARACTERISTICS OF H2 HEX BETWEEN
C           *          H2 TANK CIRCULATOR AND H2 TANK
C
      IGAS = 2
C
      WDOTX (JX,IGAS) = WDTCFH
      UCODE (JX,IGAS) = HXCODE(JX,IGAS)
      HEXHIT(JX,IGAS) = T4F21
      HEXCIT(JX,IGAS) = TKH(1)
      HEXHOT(JX,IGAS) = T5F21
      HEXCOT(JX,IGAS) = TKHMAX
      HEXCOP(JX,IGAS) = PCHFC
      HEXCIP(JX,IGAS) = PCHFC + DELTCP
C
      HSQREQ(JX,IGAS) = QMXTKH
      QTREQ(IGAS) = QTREQ(IGAS) + HSQREQ(JX,IGAS)
      CALL CSUBP(TKH(1),PCHFC,2,CPCLDF(JX,IGAS))
      TMF214 = (T4F21 + T5F21)/2.0
      CPF214 = CSPF21(TMF214)
      CPHOTF(JX,IGAS) = CPF214
      HSGCPE(JX,IGAS) = CPF214
      HSGTOT(JX,IGAS) = QMXTKH/(CPF214 * (T4F21 - T5F21))
      WDOTH(JX,IGAS) = QMXTKH/(CPF214 * (T4F21 - T5F21))
```

********   FUELCL   ********

```
C
      CALL HEXF21(IGAS,QMXTKH,HEXCIP(JX,IGAS),WHXTOT(JX,IGAS))
C
C
      CALL OTPHXF
C
C
C ***************************************************************************
C
C        *** COMPUTE THE TANK ENERGY HISTORY AND HEATER DUTY CYCLE ***
C
      TIMINC = 1.0                      @ PERIOD DEFINED BY GAS FLOW
      PSET1 = PLSET1
      PSET2 = PLSET2
      PTANK1 = PCOFC
      PTANK2 = PCHFC
C
      L1 = 0
      DO 200 K1 = 1,NDCYCL,2
      L1 = L1 + 1
      TIM(L1) = DCYCLE(K1)
  200 CONTINUE
C
      TIME = 0.0
      ICHTO = 0
      ICHTH = 0
      TKOW = 0.0
      TKHW = 0.0
      Q1CUM  = 0.0
      Q2CUM  = 0.0
C
      DO 210 K = 1,KCYCLE
C
      CTIM = TIM(K) * 60.0
      IF(CTIM.LT.10.0) GO TO 211
      NTP = (TIM(K) * 6.0) + 0.6
      WDTO30 = WDTFCO(K)/6.0
      WDTH30 = WDTFCH(K)/6.0
      GO TO 212
C
  211 NTP = CTIM + 0.6
      WDTO30 = WDTFCO(K)/60.0
      WDTH30 = WDTFCH(K)/60.0
C
  212 CONTINUE
      JP = PAGE(0)
      JP = PAGE(10)
C
      WRITE (IOT,700)
  700 FORMAT(/T42,'*** COMPUTE TANK ENERGY HISTORY AND HEATER DUTY CYCLE
     1 ***'/)
C
      WRITE (IOT,701) K, TIM(K)
  701 FORMAT(T5,'SYSTEM DUTY CYCLE NUMBER IS ',I4,T50,'CYCLE IS ',F6.2,
     1 'HOURS IN DURATION')
C
      WRITE (IOT,702) K
  702 FORMAT(T120,'CYCLE -',I2/T4,'TIME',T11,'GAS',T20,'PER-CENT',T33,'D
```

```
     1ENSITY','T44,'FLUID',T53,'THETA',T63,'PHI',T71,'Q-REQD',T81,'Q-CUM'
     2,T91,'TANK',T100,'Q-HTR.',T110,'HEATER'/T11,'FLOW',T20,'WITHDRAWN'
     3,T44,'TEMP.',T91,'PRES.',T100,'REQD.',T110,'TIME-ON',T121,'HEATER'
     4/T3,'(MIN)',T11,'(LBS)',T20,'(PERCENT)',T33,'(LB/CF)',T44,'(-R-)',
     5T53,'(B/LB)',T61,'(P-CF/B)',T72,'(BTU)',T81,'(BTU)',T91,'(PSIA)',
     6T100,'(BTU)',T110,'(MIN.)',T121,'CYCLE'/T11,'O2/H2',T22,'O2/H2',
     7T34,'O2/H2',T44,'O2/H2',T53,'O2/H2',T62,'O2/H2',T72,'O2/H2',T81,
     8'O2/H2',T91,'O2/H2',T100,'O2/H2',T110,'O2/H2',T121,'O2/H2'/)
C
C
      DO 220 I = 1,NTP
C
C     * SET TIME COUNTERS TO APPROPRIATE TIME SUB-INTERVAL *
C
      IF(I.GT.1) GO TO 226
      PTIME = CTIM + TIME
  226 CONTINUE
C
      DTIME = PTIME - TIME
      IK2 = DTIME
      IF(CTIM.LT.10.0) GO TO 225
      IF(DTIME.LT.10.0) GO TO 221
      IK2 = 0
      TIME = TIME + 10.0
      GO TO 222
C
  224 DTIME = PTIME - TIME
  221 TIME = TIME + 1.0
      IK2 = IK2 - 1
      WDTO30 = WDTFCO(K)/60.0
      WDTH30 = WDTFCH(K)/60.0
      GO TO 222
C
  225 TIME = TIME + 1.0
      IK2 = 0
C
C
  222 CONTINUE
      TKOW = TKOW + WDTO30
      TKODP = TKOW
      TKHW = TKHW + WDTH30
      TKHDP = TKHW
C
      PCOXW = TKODP/(WOCONS + 0.2 * WOCONS)
      PCH2W = TKHDP/(WHCONS + 0.2 * WHCONS)
C
C
      C3 = 1.0 - ((0.0427*PFHFC)/(ZFH*TFHFC))
      C4 = 1.0 - ((0.04253*PFOFC)/(ZFO*TFOFC))
C
      ORHO = 70.126 * (1.0 - (PCOXW * C4))
      HRHO = 4.365 * (1.0 - (PCH2W * C3))
C
      CALL FINTAB (NTBID(8))
      XTAB(1) = PTANK1
      XTAB(2) = ORHO
      OXTEM = MIPE(2,XTAB)
C
```

```
      CALL FINTAB (NTBID( 7))
      XTAB(1) = PTANK2
      XTAB(2) = HRHO
      H2TEM = MIPE(2,XTAB)
C
C
      CALL PHTHON(OXTEM,ORHO,1,PHI,THETA)
      DQDM1 = THETA
      DPDU1 = PHI
C
      CALL FINTAB (NTBID(5))
      XTAB(1) = PTANK2
      XTAB(2) = HRHO
      DQDM2 = MIPE(2,XTAB)
C
      CALL FINTAB (NTBID(40))
      XTAB(1) = PTANK2
      XTAB(2) = HRHO
      DPDU2 = MIPE(2,XTAB)
C
C
      QDTTK1 = WDTO30 * DQDM1
      QDTTK2 = WDTH30 * DQDM2
      Q1CUM  = Q1CUM  + QDTTK1
      Q2CUM  = Q2CUM  + QDTTK2
C
      CALL BETAB(OXTEM,ORHO,1,BETAO)
C
      CALL FINTAB (NTBID(46))
      XTAB(1) = PTANK2
      XTAB(2) = H2TEM
      BETAH = MIPE(2,XTAB)
C
      CALL CSUBP(OXTEM,PTANK1,1,CPO)
      CALL CSUBP(H2TEM,PTANK2,2,CPH2)
C
C
      DELP1 = TIMINC * ((DPDU1/VOLTNK(1)) * ((-CPO/BETAO)*WDTO90))
      DELP2 = TIMINC* ((DPDU2/VOLTNK(2)) * ((-CPH2/BETAH) * WDTH90))
C
      PTANK2 = PTANK2 + DELP2
      PTANK1 = PTANK1 + DELP1
C
      QHSF21 = ((QDTFC * PKW(K) * NEOP(K))/60.0) * 0.75
C
      IF(PTANK1.LE.PSET1) GO TO 130
      GO TO 135
  130 QTANK1 = Q1CUM
      HTRON1 = QTANK1/QHSF21
      ICHTO = ICHTO + 1
  135 IF(PTANK2.LE.PSET2) GO TO 140
      GO TO 118
  140 QTANK2 = Q2CUM
      HTRON2 = QTANK2/QHSF21
      ICHTH = ICHTH + 1
C
  118 CONTINUE
C
```

******** FUELCL ********

```
      IF(PAGE(2)) GO TO 227
      GO TO 228
  227 WRITE (IOT,702) K
      JP = PAGE(6)
  228 CONTINUE
C
      WRITE (IOT,703) TIME,WDTO30  ,PCOXW,ORHO,OXTEM,DQDM1,DPDU1,QDTTK1,
     1 Q1CUM,PTANK1,QTANK1,HTRON1,ICHTO,WDTH30,PCH2W,HRHO,H2TEM,DQDM2,
     2 DPDU2,QDTTK2,Q2CUM ,PTANK2,QTANK2,HTRON2,ICHTH
  703 FORMAT(T1,F9.2,T11,F6.3,T21,F6.5,T32,F8.5,T43,F6.2,T52,F6.2,T61,
     1 F6.2,T71,F6.1,T80,F7.0,T90,F7.2,T99,F7.0,T109,F7.2,T120,I6/T11,
     2 F6.3,T21,F6.5,T32,F8.5,T43,F6.2,T52,F6.2,T61,F6.2,T71,F6.1,T80,
     3 F7.0,T90,F7.2,T99,F7.0,T109,F7.2,T120,I6)
C
      IF(PTANK1,LE.PSET1) HTRON1 = 0.0
      IF(PTANK1,LE.PSET1) QTANK1 = 0.0
      IF(PTANK1,LE.PSET1) Q1CUM  = 0.0
      IF(PTANK1,LE.PSET1) PTANK1 = PCOFC
      IF(PTANK2,LE.PSET2) HTRON2 = 0.0
      IF(PTANK2,LE.PSET2) QTANK2 = 0.0
      IF(PTANK2,LE.PSET2) Q2CUM  = 0.0
      IF(PTANK2.LE.PSET2) PTANK2 = PCHFC
C
      IF(IK2.GE.1) GO TO 224
C
      IF((IK2.EQ.0).AND.(DTIME.LE.0.0)) GO TO 223
C
      IF(CTIM.LT.10.0) GO TO 229
C
      IF((DTIME.LT.10.0).AND.(DTIME.LE.0.0)) GO TO 224
C
  229 CONTINUE
C
  220 CONTINUE
C
  223 CONTINUE
C
  210 CONTINUE
C
C ***************************************************************************
C
      WRITE (IOT,6099)
 6099 FORMAT(/////T25,'***** THE SUPERCRITICAL FUEL CELL CALCULATIONS HA
     1VE BEEN COMPLETED *****')
C
      RETURN
C
      END
```

********* SUBRØUTINE GASGEN

```
      SUBROUTINE GASGEN(JX,IGAS)
C
      INCLUDE CDCYCL
      INCLUDE CHEX
      INCLUDE CHSORC
      INCLUDE CTANK
C
C        ***** COMPUTE THE HEAT SOURCE PARAMETERS.
C
      WGGFU (JX,IGAS) = 0.0
      WGGFX (JX,IGAS) = 0.0
      WGGSBT(JX,IGAS) = 0.0
      HSQREQ(JX,IGAS) = 0.0
      HSGTOT(JX,IGAS) = 0.0
      HSGCPE(JX,IGAS) = 0.0
C
      JHSTYP = HSTYPE(JX,IGAS)
C
      GO TO (10,20,30), JHSTYP
C
C        **** FOR GAS GENERATOR ONLY AS HEAT SOURCE
C
   10 HSGSUM = 0.0
      DO 40 I2 = 1,NDCYCL,2
      HSGSUM = HSGSUM + WDOTH(JX,IGAS) * DCYCLE(I2)
   40 CONTINUE
C
      WGGFX(JX,IGAS) = HSGSUM - WGGFU(JX,IGAS)
      WGGSBT(JX,IGAS) = WGGSBT(JX,IGAS) + WGGFX(JX,IGAS)
      WGGTOT(IGAS) = WGGTOT(IGAS) + WGGFX(JX,IGAS)
C
      ATERM = 13.824204 - (0.0117823*HSPRES(JX,IGAS)) + (1.8632927E-5 *
     1(HSPRES(JX,IGAS)**2)) - (1.108423E-8 * (HSPRES(JX,IGAS)**3))
C
      BTERM = 7.9470262 - (.035636198*HSPRES(JX,IGAS)) + (6.4684644E-5 *
     1(HSPRES(JX,IGAS)**2)) - (3.7946E-8 * (HSPRES(JX,IGAS)**3))
C
      HSWGHT(JX,IGAS) = ATERM + BTERM * WDOTH(JX,IGAS)
C
      HXASSY(JX,IGAS) = WHXTOT(JX,IGAS) + HSWGHT(JX,IGAS)
C
      RETURN
C
C        ***** FOR WASTE HEAT ONLY AS HEAT SOURCE.
C
   20 HSGSUM = 0.0
      DO 50 I2 = 1,NDCYCL,2
      HSGSUM = HSGSUM + WDOTH(JX,IGAS) * DCYCLE(I2)
   50 CONTINUE
C
      CALL CSUBP1(HSOTEM(JX,IGAS),HSMRAT(JX,IGAS),CPEG)
      HSQREQ(JX,IGAS) = HSGSUM * CPEG*(HEXHIT(JX,IGAS)-HEXHOT(JX,IGAS))
      QTREQ(IGAS) = QTREQ(IGAS) + HSQREQ(JX,IGAS)
      HSGTOT(JX,IGAS) = HSGSUM
      HFTOT(IGAS) = HFTOT(IGAS) + HSGTOT(JX,IGAS)
      HSGCPE(JX,IGAS) = CPEG
C
```

```
            RETURN
      C
      C        ***** FOR GAS GENERATOR AND WASTE HEAT BOTH AS HEAT SOURCES.
      C
         30 WGGFU(JX,IGAS) = HSAEE(JX,IGAS)/((3.11625 + HSMRAT(JX,IGAS) *
           1 (0.46875 * HSMRAT(JX,IGAS) - 1.675)) * (HEXHIT(JX,IGAS) -
           2 HEXHOT(JX,IGAS)))
      C
            GO TO 10
      C
            END
```

********* SUBRØUTINE GETCØN

```
C                   * * * * * * * * * * * * * * * * * * * * * * * * *
C                   * ROUTINE NAME - ROUTINE TO UNPACK THE FIRST    *
C                   *                WORD OF THE CONFIGURATION      *
C                   *                TABE. SEE S.R. STOCON FOR      *
C                   *                WORD FORMAT                    *
C                   * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                   * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                   * DATE CODED   - 3/11/70                        *
C                   * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE GETCON(IDX)
C
C         ***** EXPLANATION OF THE CALLING SEQUENCE
C               *
C         ***** IDX - INDEX OF THE CONFIGURATION TABLE
C
      LOGICAL DIAG
C
      INCLUDE CCNFIG
      INCLUDE CIOUNT
C
      DO 10 II = 1,6
      ICNFIG(II) = IPBYTE(CONFIG(IDX,1),II,0,6)
   10 CONTINUE
C
      IF (DIAG(1,6HGETCON)) WRITE (IOT,6000) IDX,ICNFIG
      RETURN
 6000 FORMAT ('+'14X,7I5)
      END
```

******** FUNCTIØN GØMTRY

```
C
      FUNCTION CONE (R,H)
C
      INCLUDE CONST,LIST
C
C                     VOLUME OF CONE (CIRCULAR)
      CONE = PI*R*R*H / 3.0
      RETURN
C                     VOLUME OF CYLINDER (RIGHT-CIRCULAR)
      ENTRY CYLNDR (R,H)
      CONE = PI*R*R*H
      RETURN
C                     VOLUME BETWEEN CYLINDER AND SPHEROID
C                     RROT IS ALONG AXIS OF ROTATION
      ENTRY CYLSPH (RROT,R)
      CONE = PI*R*R*RROT / 3.0
      RETURN
C                     VOLUME OF FRUSTRUM OF CONE (CIRCULAR)
      ENTRY FRCONE (R,H,R2)
      CONE = PI*H*(R*R + R2*R2 + R*R2) / 3.0
      RETURN
C                     VOLUME OF HEMISPHERE OR HALF OF SPHEROID
C                     RROT IS ON AXIS OF ROTATION
      ENTRY HSPHER (RROT,R)
      CONE = PI203*R*R*RROT
      RETURN
C                     VOLUME OF SPHERE OR SPHEROID
C                     RROT IS ALONG AXIS OF ROTATION
      ENTRY SPHERE (RROT,R)
      CONE = 2.0*PI203*R*R*RROT
      RETURN
C
C                     AREA OF CYLINDER
      ENTRY ARACYL (R,H)
      CONE = 2.0*PI*R*H
      RETURN
C
C                     AREA OF FRUSTRUM
      ENTRY AREAFR (R,H,R2)
      CONE = PI*(R+R2)*SQRT (H*H+(R-R2)**2)
      RETURN
C
C                     AREA OF HALF OF SPHEROID
C                     RROT ALONG AXIS OF ROTATION
      ENTRY ARSPHR (RROT,R)
      IF (RROT .LE. R) GO TO 110
C                     ROTARED ABOUT MAJOR AXIS
C                     E = ECCENTRICITY FOR ELLIPSE
      E = SQRT (RROT*RROT - R*R)/RROT
      CONE = PI*R*(R+RROT*ASIN(E)/E)
      RETURN
  110 IF (RROT .EQ. R) GO TO 120
C                     ROTARED ABOUT MINOR AXIS
      E = SQRT (R*R - RROT*RROT) / R
      CONE = PI*(R*R+(RROT*RROT/(2.*E))*ALOG((1.+E)/(1.-E)))
      RETURN
C                     AREA OF HEMISPHERE
```

```
********    GOMTRY    ********

      120 CONE = 2.*PI*R*R
          RETURN
          END
```

******** SUBRØUTINE HEATEX

```
C
C                              * * * * * * * * * * * * * * * * * * *
C                              * HEAT EXCHANGER DESIGN ROUTINE *
C                              * UNIVAC 1108           FORTRAN  V *
C                              * DATE CODED          JULY 27,1971 *
C                              * J. MCKAY      BLDG 201  X45178 *
C                              * REVISION 1              JUNE 1972 *
C                              * REVISION 2            AUGUST 1972 *
C                              * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE HEATEX(IGAS,JHEX,WDOTCI,THIN,TCIN,THOUT,TCOUT,PHIN,
     1                   PCIN,PHOUT,PCOUT,OFR,WDOTHI,TOTHHX)
C
      REAL ICIN,ICOUT,NTU
C
      INCLUDE CIOUNT
      INCLUDE CHEX
      INCLUDE TABLOK
C
      DIMENSION F1(2),F2(2),F3(2),F4(2),F5(2)
C
      DATA  F1/700.,180./ F2/160.,35./ F3/278.2,59.4/ F4/731.4,187.5/
     1  F5/500.,350./
C
 6010 FORMAT ('0'T10,'**** SPECIFY COLD FLUID TYPE ****')
 6020 FORMAT ('0'T10,'*****  ERROR TERMINATION IN HEATEX  *****')
 6030 FORMAT ('0'T10,'*** THERMAL BYPASS REQUIRED ***')
 6040 FORMAT ('0'T10,'*** T-HOT(OUT) > T-HOT(IN) ***')
 6050 FORMAT ('0'T10,'*** T-HOT(IN) MUST BE < 3500 DEG-R ***')
 6060 FORMAT ('0'//T10,'*** TCR ='F7.3,' OUT OF RANGE -- HEX. NO.'I3,
     1 '  GAS TYPE'I2,'  SUBUNIT'I2,' ***')
 6080 FORMAT ('0'T10,'*** PCIN ='F9.2,' NOT IN RANGE ***')
 6090 FORMAT ('0'T10,'*** TCIN ='F9.2,'  <  'F9.2,' ***')
 6100 FORMAT ('0'T10,'*** TCOUT ='F9.2,'  <  ' F9.2,' ***')
C
C
      DO 3000 JKM=1,3
      NCVG = 0
C
C     INITIALIZE OUTPUT VARIABLES
C
      DO 12 I = 1,3
      TCRU  (JHEX,I,IGAS) = 0.0
      FWDTHU(JHEX,I,IGAS) = 0.0
      FDPCU (JHEX,I,IGAS) = 0.0
      CRU   (JHEX,I,IGAS) = 0.0
      NTUU  (JHEX,I,IGAS) = 0.0
      UAU   (JHEX,I,IGAS) = 0.0
      WOUAU (JHEX,I,IGAS) = 0.0
      WTHXU (JHEX,I,IGAS) = 0.0
   12 CONTINUE
C
C        *** THE O/F RATIO IS RESTRICTED TO ONE, PCOUT AND PHOUT ***
C        *** MUST BE SPECIFIED                                   ***
C
      OF = 1.
      WDOTC = WDOTCI
```

```
********      HEATEX    ********

          WDOTHI = 0.
      C                             IGAS = 1 OXYGEN , = 2 HYDROGEN
          IF (IGAS .GT. 0 .AND. IGAS .LT. 3) GO TO 110
          WRITE (IOT,6010)
          RETURN
      110 CONTINUE
      C                   CALC. PRESSURE DROPS
          DPH = PHIN - PHOUT
          DPC = PCIN - PCOUT
      C                   CHECK FOR ACCEPTABLE PRESSURE DROPS
          IF (DPH .LE. 0.4*PHIN) GO TO 120
          DPH = 0.4*PHIN
          PHOUT = PHIN - DPH
      120 IF (DPC .LE. 0.25*PCOUT) GO TO 130
          DPC = 0.25 * PCOUT
          PCIN = PCOUT + DPC
      130 CONTINUE
      C                             CALC. TEMPERATURE DROPS
          DTC = TCOUT - TCIN
          DTH = THIN - THOUT
          DHCIN = THIN - TCIN
      C
          FWDP = 1.0
          FWOF = 1.0
      C                             CALCULATE ENTHALPY OF COLD FLUID
          CALL ENTHOH (PCIN,TCIN,IGAS,ICIN)
          CALL ENTHOH (PCOUT,TCOUT,IGAS,ICOUT)
      C                             FIND CHANGE IN ENTHALPY (COLD FLUID)
          DIC = ICOUT - ICIN
      C                             CALC. P(SAT)
          PSAT = .126 * PHOUT * OF
      C                             LOOKUP T(SAT)
          CALL FINTAB (NTBID(23))
          THSAT = MIPE (1,PSAT)
      C                             CHECK FOR POSSIBLE CONDENSATION
          IF (THOUT .GT. THSAT+30.) GO TO 230
          THOUT = THSAT + 30.
          DTH = THIN - THOUT
      230 IF (THOUT .LT. THIN) GO TO 240
          WRITE (IOT,6040)
          GO TO 4000
      240 IF (THIN .LE. 3500.) GO TO 250
          WRITE (IOT,6050)
          GO TO 4000
      250 CONTINUE
      C                             CALCULATE EFFECTIVENESS
          ISW = 1
          TCOSAV = TCOUT
          EFFC = DTC / DHCIN
      260 EFFH = DTH / DHCIN
      C                             CALC. EFFECTIVENESS SUM
          EFSUM= EFFC + EFFH
      C
      C                   CHECK RANGE OF EFFECT. SUM
          IF (EFSUM .GE. 0.9) GO TO 300
          GO TO (270,280),ISW
      270 THOUT = THSAT + 30.
          DTH = THIN - THOUT
```

```
      ISW = 2
      GO TO 260
C                    IF EFSUM CAN'T BE RAISED BY THOUT, THEN CHANGE THE
C                    PARTIAL TCOUT BY USING A BYPASS
  280 EFSUM = 0.5
      XTCO = THOUT - DHCIN/2.0
C                    THERMAL BYPASS
      WDOTC = WDOTC*DTC / (XTCO-TCIN)
      TCOUT = XTCO
      DTC = TCOUT - TCIN
      WRITE (IOT,6030)
      CALL ENTHOH (PCOUT,TCOUT,IGAS,ICOUT)
      DIC = ICOUT - ICIN
  300 IF (EFSUM .LE. 0.91) GO TO 310
      EFSUM = 0.91
      THOUT = DTC + THIN - EFSUM*DHCIN
      DTH  = THIN - THOUT
  310 CONTINUE
C                              OF INPUT -- LOOKUP C SUB P (COMB.PROD.)
      ISW2 = 1
      CALL FINTAB (NTBID(24))
  312 XTAB(1) = (THOUT + THIN) / 2.0
      XTAB(2) = OF
      ACSBPH = MIPE (2,XTAB)
      DIH = ACSBPH*DTH
      WDOTHI = WDOTC*DIC/DIH
      ACSBPC = DIC / DTC
      CC = WDOTC*ACSBPC
      CH = WDOTHI*ACSBPH
      CMIN = CC
      IF (CMIN .GT. CH) CMIN = CH
C                    FIND EFFECTIVENESS
      EFFT = CC*DTC / (CMIN*DHCIN)
      IF (EFFT .LT. 0.9) GO TO 320
      GO TO (314,314,320),ISW2
  314 ISW2 = ISW2 + 1
      EFFT = 0.895
      THOUT = THIN - EFFT*CMIN / CH*DHCIN
      DTH = THIN - THOUT
      GO TO 312
  320 TCSAT = TSAT (PCOUT,IGAS)
C
C          *** SUBDIVIDE HEAT EXCH. ACCORDING TO HEAT TRANSFER RANGE ***
C
      IF (PCIN .GE. 10.0 .AND. PCIN .LE. 2250.) GO TO 340
      WRITE (IOT,6080) PCIN
      GO TO 4000
  340 IF (PCIN .GT. F1(IGAS)) GO TO 370
      IF (TCIN .LT. F2(IGAS) .OR. TCIN .GT. TCSAT) GO TO 400
      IF (ABS(TCOUT-TCSAT) .GT. 0.005*TCOUT) GO TO 350
C                    HAVE BOILING SUB-UNIT ONLY
      ISU = 8
      GO TO 460
  350 IF (TCOUT .LE. 500.) GO TO 360
C                    HAVE BOILING, PARALLEL, COUNTER-FLOW SUBUNITS
      ISU = 11
      GO TO 460
C                    HAVE BOILING AND PARALLEL-FLOW SUB-UNITS
```

```
  360 ISU = 10
      GO TO 460
  370 IF (TCIN .LT. F2(IGAS) .OR. TCIN .GE. F3(IGAS)) GO TO 400
      IF (TCOUT .GT. F3(IGAS)) GO TO 380
C                     HAVE SUPERCRITICAL SUB-UNIT ONLY
      ISU = 4
      GO TO 460
  380 IF (TCOUT .LE. 500.) GO TO 390
C                     HAVE SUPERCRITICAL, PARALLEL, COUNTER-FLOW SUB-UNITS
      ISU = 7
      GO TO 460
C                     HAVE SUPERCRITICAL AND PARALLEL FLOW SUBUNITS
  390 ISU = 6
      GO TO 460
  400 IF (TCIN .LT. 500.) GO TO 410
C                     HAVE COUNTER FLOW SUBUNIT ONLY
      ISU = 1
      GO TO 460
  410 IF (PCIN .GE. F4(IGAS)) GO TO 420
      IF (TCIN .GE. TCSAT) GO TO 430
      WRITE (IOT,6090) TCIN,TCSAT
      GO TO 4000
  420 IF (TCIN .GE. F3(IGAS)) GO TO 430
      WRITE (IOT,6090) TCIN,F3(IGAS)
      GO TO 4000
  430 IF (TCOUT .LE. 500.) GO TO 440
C                     HAVE PARALLEL AND COUNTER FLOW SUBUNITS
      ISU = 3
      GO TO 460
  440 IF (TCOUT .GT. F3(IGAS)) GO TO 450
      WRITE (IOT,6100) TCOUT,F3(IGAS)
      GO TO 4000
C                     HAVE PARALLEL FLOW SUBUNIT ONLY
  450 ISU = 2
C                     HEAT EXCHANGER NOW SUB-DIVIDED
  460 CONTINUE
C
      NSU = 0
      DO 470 I=4,1,-1
      NKU = 2**(I-1)
      IF (AND(ISU,NKU) .NE. NKU) GO TO 470
      NSU = NSU + 1
      NSS = I
  470 CONTINUE
      IF (NSS .EQ. 4) NSS = 3
      NST = 4 - NSS
      NSS = NST + 1 - NSU
C
C         *** NOW GO THROUGH REQUIRED SUBUNITS AND COMPUTE PARAMETERS ***
C
      FPCIN = PCIN
      FTCIN = TCIN
      SDPC = 0.
      TOTWHX = 0.
C
      DO 2000 IS=NSS,NST
C                     THREE POSSIBLE SUBUNITS
C                     SET TEMP., AND PRES. LIMITS EACH UNIT
```

```
*********    HEATEX    *********

            GO TO (480,530,570),IS
      C                    BOILING OR SUPERCRITICAL
        480 IF (ISU .LE. 7) GO TO 490
      C                    BOILING
            FTCOUT = TSAT (PCIN-DPC/2.,IGAS)
            GO TO 590
      C                    SUPERCRITICAL
        490 FTCOUT = TCOUT
            IF (TCOUT .GT. F3(IGAS)) FTCOUT = F3(IGAS)
            GO TO 590
      C                    PARALLEL FLOW UNIT
        530 IF (ISU .LE. 3) GO TO 540
            FTCIN = FTCOUT
            FPCIN = FPCOUT
        540 FTCOUT = TCOUT
            IF (AND(ISU,1) .NE. 0) FTCOUT = 500.
            GO TO 590
      C                    COUNTER FLOW UNIT
        570 IF (ISU .EQ. 1) GO TO 580
            FTCIN = FTCOUT
            FPCIN = FPCOUT
        580 FTCOUT = TCOUT
        590 CALL ENTHOH (FPCIN,FTCIN,IGAS,FIIN)
            IF (DPC .GT. 0.) GO TO 600
            FDPC = 0.2*FPCIN / NSU
            GO TO 610
        600 CALL ENTHOH (FPCIN-DPC/2.,FTCOUT,IGAS,FIOT)
            FDPC = DPC / DIC * (FIOT-FIIN)
        610 CALL ENTHOH (FPCIN-FDPC,FTCOUT,IGAS,FIOT)
            FDIC = FIOT-FIIN
            FDPC = DPC / DIC * FDIC
            IF (IS .EQ. 1) FDPC = FDPC/2.
            FPCOUT = FPCIN - FDPC
            FWDTH = WDOTC*FDIC/DIH
      C                    PRES - TEMP SUBUNIT RANGES HAVE BEEN SET
            IERR = 0
            IF (IS .NE. 3) GO TO 630
      C                    COUNTER FLOW
            CALL TCRCAL (THIN,THOUT,FTCOUT,FTCIN,TCRMIN,TCRMAX,IERR)
            GO TO 640
      C                    SUPERCRITICAL, BOILING OR PARALLEL
        630 CALL TCRCAL (THIN,THOUT,FTCIN,FTCOUT,TCRMIN,TCRMAX,IERR)
        640 IF (IERR .NE. 0) GO TO 4000
      C                    DETERMINE NUMBER OF HEAT TRANSFER UNITS
            FDTC = FTCOUT - FTCIN
            FEMX = DTH
            IF (FDTC .GT. DTH) FEMX = FDTC
            FEMX = FEMX/(THIN-FTCIN)
            FDIH = WDOTC*FDIC/FWDTH
            AVCPC = FDIC/FDTC
            AVCPH = FDIH/DTH
            CMIN = WDOTC*AVCPC*3600.
            CMAX = FWDTH*AVCPH*3600.
            CR = CMIN
            IF (CMAX .GE. CMIN) GO TO 670
            CR = CMAX
            CMAX = CMIN
            CMIN = CR
```

```
      670 CR = CR/CMAX
          GO TO (680,690,700),IS
      680 IF (ISU .LE. 7) GO TO 690
C                                BOILING
          NTU = -ALOG (1.0-FEMX)
          GO TO 710
C                                SUPERCRITICAL OR PARALLEL
      690 NTU = -ALOG (1.-FEMX-CR*FEMX) / (1.0+CR)
          GO TO 710
C                                COUNTER FLOW
      700 NTU = ALOG ((1.-FEMX)/(1.-CR*FEMX)) / (CR-1.)
C                                CALC.  UA FOR EACH UNIT
      710 UA = NTU*CMIN
          IF (IS .GT. 1) GO TO 720
          IF (ISU .GT. 7) GO TO 720
C                                CALCULATION FOR SUPERCRITICAL ONLY
          GDPC = FDPC*F5(IGAS) / FTCOUT
          CALL TCRCLC  (IGAS,GDPC,FPCIN,DPH,PHIN,TCR)
          IF (TCR .GT. 10.0) TCR = 10.0
          CALL WOUACL (WOUA)
C                                SPECIAL FORM OF WEIGHT CALC. FOR SUPERCRITICAL
          WTHX = WOUA*UA*(1.+TCR)/(1.+TCRMAX)*TCRMAX/TCR
          GO TO 740
C
C                                CALC. TCR FOR O2 OR H2
      720 CALL TCRCLC  (IGAS,FDPC,FPCIN,DPH,PHIN,TCR)
C                                IS TCR WITHIN ACCEPTABLE RANGE
      721 IF (TCR .LE. TCRMAX) GO TO 724
C                                MUST LOWER TCR
C                                LOWER  (DPC/PCIN)
          CALL TCRLOW (FPCOUT,IGO)
          GO TO (721,722),IGO
C                                TRY RAISING  (DPH/PHIN)
      722 DPH = 2.*DPH
          IF (DPH .LE. .4*PHIN) GO TO 723
          PHIN = PHOUT / 0.6
          DPH = 0.4*PHIN
          CALL TCRCLC  (IGAS,FDPC,FPCIN,DPH,PHIN,TCR)
          IF (TCR .LE. TCRMAX) GO TO 730
          NCVG = IS
          ETCR = TCR
          GO TO 730
      723 PHIN = PHOUT + DPH
          CALL TCRCLC  (IGAS,FDPC,FPCIN,DPH,PHIN,TCR)
          IF (TCR .GT. TCRMAX) GO TO 722
C                                CHECK FOR TCR TOO LOW
      724 IF (TCR .GE. TCRMIN) GO TO 730
C                                RAISE  (DPC/PCIN)
          CALL TCRRAZ (FPCOUT,IGO)
          GO TO (724,725),IGO
      725 CONTINUE
          NCVG = IS
          ETCR = TCR
C                                CALC.  (W/UA)
      730 CALL WOUACL (WOUA)
C                                CALCULATE WEIGHT FOR THIS SUB-UNIT
          WTHX = WOUA * UA
C                                TOTAL HEAT EXCHANGER WEIGHT
```

```
  740 TOTWHX = TOTWHX + WTHX
C
      SDPC = SDPC + FDPC
C
      TCRU  (JHEX,IS,IGAS) = TCR
      FWDTHU(JHEX,IS,IGAS) = FWDTH
      FDPCU (JHEX,IS,IGAS) = FDPC
      CRU   (JHEX,IS,IGAS) = CR
      NTUU  (JHEX,IS,IGAS) = NTU
      UAU   (JHEX,IS,IGAS) = UA
      WOUAU (JHEX,IS,IGAS) = WOUA
      WTHXU (JHEX,IS,IGAS) = WTHX
C
 2000 CONTINUE
      PCIN = PCOUT + SDPC
      TCOUT = TCOSAV
      IF (NCVG .EQ. 0) GO TO 4010
 3000 CONTINUE
      WRITE (IOT,6060) ETCR,JHEX,IGAS,NCVG
      GO TO 4010
 4000 CONTINUE
      WRITE (IOT,6020)
      CALL EXIT
C
 4010 CONTINUE
      CPCLDF(JHEX,IGAS) = ACSBPC
      CPHOTF(JHEX,IGAS) = ACSBPH
      EPSLNC(JHEX,IGAS) = EFFC
      EPSLNH(JHEX,IGAS) = EFFH
      EPSLNS(JHEX,IGAS) = EFSUM
      IISU(JHEX,IGAS) = ISU
      NSSK(JHEX,IGAS) = NSS
      NSUK(JHEX,IGAS) = NSU
      IF (TOTWHX .LT. 5.0) TOTWHX = 5.0
C
      RETURN
C
      END
```

******** SUBRØUTINE HEXELC

```
      C
      C  **********************************************************************
      C
      C     THE SUBROUTINE COMPUTES THE WEIGHT AND DELTAP FOR ELECTRICALLY
      C     HEATED HEAT EXCHANGERS EMPLOYING THE PROCEDURES SET FORTH IN
      C     AR-DES.REF.MAN.-EXT.PRESSN.SYS.FOR CRYO.STOR.SYSTEMS,AR-71-7535,
      C     SEPT.10,1971, NAS9-10453 (NASA-HOUSTON).
      C
      C          **** NOTE THAT PROGRAM ASSUMES THE HEATER FLUX (HF=(Q/A)REF.)
      C             * TO BE RATED AT A REFERENCE TEMPERATURE OF 360 DEGREES
      C             * RANKINE. (SEE AR-71-7537,SEC.5) IF A DIFFERENT REFERENCE
      C             * TEMPERATURE IS TO BE USED, THEN 'TREF' MUST BE CHANGED.
      C          **** A NI-FE ALLOY RESISTANCE HEATER ELEMENT IS ASSUMED.
      C
      C     PROGRAMMED BY - R.F.HAUSMAN, DEPT 62-13, 104, LMSC, 2-19-73
      C
      C  **********************************************************************
      C
            SUBROUTINE HEXELC(NGAS,TIN,TOUT,PIN,HF,LDIA,WDOT,RHOGAS,IFIN,
           1                  HEXWGT, DELTAP, UOA, DH, HLNGTH)
      C
            REAL MASVEL, LDIA
      C
            INCLUDE CIOUNT
            INCLUDE TABLOK
      C
            DH = LDIA * 2.0
            T1 = TIN
            T2 = TOUT
            PI = 3.141593
            TREF = 360.0
      C
            PC1 = 736.9                                  @ OXYGEN    CRIT.PRES.
            PC2 = 187.5                                  @ HYDROGEN  CRIT.PRES.
            PC3 = 492.2                                  @ NITROGEN  CRIT.PRES.
      C
      C          *** CALCULATE THE MASS VELOCITY
      C
         10 CONTINUE
      C
            MASVEL = WDOT/(PI*DH)
      C
      C          *** CALCULATE THE OVERALL HEAT TRANSER COEFFICIENT -U-
      C
            IF(NGAS.EQ.1)  KGS = 2
            IF(NGAS.EQ.2)  KGS = 1
            IF(NGAS.EQ.18) KGS = 2
            CALL FINTAB (NTBID(41)+KGS)
            XTAB(1) = PIN
            XTAB(2) = MASVEL
            UOA = MIPE(2,XTAB)
      C
      C          *** CALCULATE THE VARIABLE VALUES FOR HEX LENGTH EQUATION
      C
            TMEAN = (T2 + T1)/2.0
            CALL CSUBP(TMEAN,PIN,NGAS,CPBAR)
            BONE = PI * DH * HF * TREF
```

```
*********     HEXELC     *********
          BPID = 2.0 * (BONE/(PI*DH))
          PHIONE = (((T2**2)-(T1**2))/BPID)
          PHITWO = (ALOG(T2/T1))/UOA
C
C         *** COMPUTE CORRECTION FACTOR BETA FOR PHITWO
C
          IF(NGAS.EQ.1) POPC = PIN/PC1
          IF(NGAS.EQ.2) POPC = PIN/PC2
          IF(NGAS.EQ.18) POPC = PIN/PC3
C
          CALL FINTAB (NTBID(44))
          XTAB(1) = POPC
          BETA = MIPE(1,XTAB)
C
C         *** EVALUATE HEX-LENGTH EQUATION
C
          HLNGTH = MASVEL * CPBAR * (PHIONE + BETA*PHITWO)
C
C
   20     CONTINUE
C
C         *** COMPUTE  HEX WEIGHT
C
          IF(NGAS.EQ.1) GO TO 30
          IF(NGAS.EQ.2) GO TO 40
          IF(NGAS.EQ.18) GO TO 40
   30     HEXWGT = 0.1519984 * (DH**1.05379) * HLNGTH
          GO TO 50
   40     HEXWGT = 0.0950445 * (DH**1.061) * HLNGTH
   50     CONTINUE
C
C         *** IF ANTI-BURNOUT FINS ARE USED - COMPUTE FIN WEIGHT
C
          IF(IFIN.EQ.0) GO TO 60
C
          FINWGT = 0.20688721 * (DH**3.19204) * HLNGTH
C
C         *** COMPUTE  HEX TOTAL WEIGHT
C
          HEXWGT = HEXWGT + FINWGT
C
   60     CONTINUE
C
C         *** COMPUTE  HEX DELTAP FOR FLUID CONSIDERED
C
          CALL FINTAB (NTBID(45))
          XTAB(1) = HLNGTH
          XTAB(2) = MASVEL
          SIGDLP  = MIPE(2,XTAB)
C
          DELTAP  = SIGDLP/(RHOGAS/0.0765)
C
          RETURN
C
          END
```

********* SUBRØUTINE HEXF21

```
      SUBROUTINE HEXF21(IGAS,QREQ,PCIN,HXWT)
C
C     *** DATA CONTAINED IN THIS SUBROUTINE CAME FROM THE FOLLOWING
C       * SOURCE,  FREON-21 CRYOGENIC HEAT EXCHANGER PARAMETRIC DATA,-
C     *** AIRESEARCH REPORT NO, 71-7720, DTD. 9-22-71.
C
      IF(IGAS.EQ.2) GO TO 70
C
      IF(QREQ.LE.50000.0.AND.QREQ.GE.10000.0.AND.PCIN.GE.600.) GO TO 10
C
      IF(QREQ.LT.50000.0.AND.QREQ.GE.10000.0.AND.PCIN.LT.600.) GO TO 20
C
      IF(QREQ.LT.10000.0.AND.QREQ.GE.  700.0.AND.PCIN.GE.600.) GO TO 30
C
      IF(QREQ.LT.10000.0.AND.QREQ.GE.  700.0.AND.PCIN.LT.600.) GO TO 40
C
      IF(QREQ.LE.70000.0.AND.QREQ.GT.50000.0.AND.PCIN.GE.600.) GO TO 50
C
      IF(QREQ.LE.70000.0.AND.QREQ.GT.50000.0.AND.PCIN.LT.600.) GO TO 60
C
   10 IF(QREQ.LE.50000.0.AND.QREQ.GE.40000.0) HXWT = 0.80
      IF(QREQ.LT.40000.0.AND.QREQ.GE.30000.0) HXWT = 0.65
      IF(QREQ.LT.30000.0.AND.QREQ.GE.20000.0) HXWT = 0.45
      IF(QREQ.LT.20000.0.AND.QREQ.GE.10000.0) HXWT = 0.25
      RETURN
C
   20 IF(QREQ.LE.50000.0.AND.QREQ.GE.40000.0) HXWT = 0.60
      IF(QREQ.LT.40000.0.AND.QREQ.GE.30000.0) HXWT = 0.40
      IF(QREQ.LT.30000.0.AND.QREQ.GE.20000.0) HXWT = 0.25
      IF(QREQ.LT.20000.0.AND.QREQ.GE.10000.0) HXWT = 0.15
      RETURN
   30 IF(QREQ.LT.10000.0.AND.QREQ.GE.2000.0) HXWT = 0.30
      IF(QREQ.LT. 2000.0.AND.QREQ.GE. 700.0) HXWT = 0.10
      RETURN
   40 IF(QREQ.LT.10000.0.AND.QREQ.GE.2000.0) HXWT = 0.20
      IF(QREQ.LT. 2000.0.AND.QREQ.GE. 700.0) HXWT = 0.08
      RETURN
   50 IF(QREQ.LE.70000.0.AND.QREQ.GT.50000.0) HXWT = 1.0
      RETURN
   60 IF(QREQ.LE.70000.0.AND.QREQ.GT.50000.0) HXWT = 0.8
      RETURN
C
   70 IF(QREQ.LE.60000.0.AND.QREQ.GE.50000.0) HXWT = 0.7
      IF(QREQ.LT.50000.0.AND.QREQ.GE.40000.0) HXWT = 0.8
      IF(QREQ.LT.40000.0.AND.QREQ.GE.30000.0) HXWT = 0.65
      IF(QREQ.LT.30000.0.AND.QREQ.GE.20000.0) HXWT = 0.50
      IF(QREQ.LT.20000.0.AND.QREQ.GE.10000.0) HXWT = 0.40
      IF(QREQ.LT.10000.0.AND.QREQ.GE. 2000.0) HXWT = 0.25
      IF(QREQ.LT.2000.0.AND.QREQ.GE.50.0) HXWT = 0.15
C
      RETURN
C
      END
```

********   FUNCTIØN HPTCP

```
      FUNCTION HPTCP(PRES,TEMP)                          1291 0001
      HPTCP=PTHEAT(PRES,TEMP,1)                          1292 0002
      RETURN                                             1293 0003
      END                                                1294 0004
```

********* FUNCTIØN HPTCV

```
      FUNCTION HPTCV(PRES,TEMP)                                  1295 0001
      HPTCV=PTHEAT(PRES,TEMP,2)                                  1296 0002
      RETURN                                                     1297 0003
      END                                                        1298 0004
```

********* FUNCTIØN HPTGAM

```
      FUNCTION HPTGAM(PRES,TEMP)                    1299 0001
      HPTGAM = PTHEAT(PRES,TEMP,3)                  1300 0002
      RETURN                                        1301 0003
      END                                           1302 0004
```

******** FUNCTIØN HPW

```
C                  * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                  * ROUTINE NAME - HEX PROPELANT WEIGHT CALC.          *
C                  * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2        *
C                  * PROGRAMMER   - R. BOLLINGER 1943 102 26933         *
C                  * DATE CODED   - 4/21/70                             *
C                  * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION HPW(JX,IGAS)
      HPW = 0.
      RETURN
      END
```

******** SUBRØUTINE HTLEAK

```
      SUBROUTINE HTLEAK(TENV,TFLO,TFLH,WSVH)
C
C     *** THIS IS CURRENTLY A DUMMY ROUTINE
C
      WSVH = 1.0000
C
      RETURN
      END
```

******** FUNCTIØN HVAP

```
      FUNCTION HVAP(N,T,Z)
      DIMENSION HVBP(17),R(17)
      DATA(HVBP(I),I=1,17)/91.62,195.9,71.5,195.9,75.,219.2,81.9,219.2,
     1 81.9,223.,71.5,589.3,178.1,425.8,123.,208.,-1./
      DATA(R(I),I=1,17)/48.31,766.8,40.67,766.8,42.01,96.35,28.62,96.35
     1 ,28.62,55.81,40.67,90.77,16.8,36.95,11.9,33.5,386.3/
      IF(N.EQ.2.OR.N.EQ.4) GO TO 5
      HVAP=HVBP(N)*((1.-T/TBOIL(N))/(1.-TSTART(N)/TBOIL(N)))**.38
     1  -R(N)*T*Z/778.3
      RETURN
    5 HVAP=-160.05404+36,424565*T-1.41762691*T*T+.026014121*T*T*T-
     1 .20992651E-3*T**4+.3154429E-6*T**5
      RETURN
      END
```

******** FUNCTIØN HYENTH

```
C
      FUNCTION HYENTH(PRES,TEMP)                                        0001 0002
C
      DIMENSION LOC(29),JP(24),MX(24),BP(24),DP(24),BT(29),DT(24),PS(20) 0002 0004
     1,TS(20),TL(10),HS(10), H(1183)                                    0003 0005
      DIMENSION AA( 93),AB(100),AC(110),AD(101),AE(101),AF(113),AG(103) 0004 0006
     1 ,AH(112),AI(106),AJ(119),AK(104),AL( 21)                         0005 0007
C
      EQUIVALENCE( H,AA),( H(  94),AB),( H( 194),AC),( H( 304),AD)      0006 0009
     1    ,( H( 405),AE),( H( 506),AF),( H( 619),AG),( H( 722),AH)      0007 0010
     2    ,( H( 834),AI),( H( 940),AJ),( H(1059),AK),( H(1163),AL)      0008 0011
C
      DATA LOC/1,16,31,46,90,145,157,177,195,231,261,279,306,378,441,468 0009 0013
     1,558,612,642,682,847,892,1028,1073,1103,1112,1121,1130,1160/      0010 0014
      DATA JP/3,3,3,4,5,2,5,3,6,5,3,3,6,9,9,9,9,5,5,11,9,17,5,5/        0011 0015
      DATA MX/1,1,1,2,3,0,3,1,4,3,1,1,4,7,7,7,7,3,3,9,7,15,3,3/         0012 0016
      DATA BP/0.,-100.,0.,0.,.9999999,0.,300.,2645.28,1175.68,587.84,0.,0013 0017
     12645.28,1175.68,0.,587.84,587.84,293.92,0.,440.88,293.92,176.352,1 0014 0018
     276.352,0.,117.568/                                                0015 0019
      DATA DP/1500.,300.,40.,10.,1.,5000.,1100.,1175.68,293.92,146.96,  0016 0020
     1293.92,1175.68,293.92,146.96,73.48,73.48,36.74,73.48,36.74,14.696, 0017 0021
     214.696,7.348,44.088,14.696/                                       0018 0022
      DATA BT/3000.,3000.,3000.,3000.,3000.,500.,170.,90.,90.,90.,90.,  0019 0023
     132.4,30.6,27.,75.6,59.4,72.,72.,59.4,59.4,64.8,59.4,28.8,54.,     0020 0024
     25000.,5000.,5000.,5000.,5000./                                    0021 0025
      DATA DT/500.,500.,500.,200.,200.,500.,110.,18.,18.,18.,18.,7.2,5.4 0022 0026
     1,5.4,7.2,1.8,3.6,3.6,1.8,.9,1.8,.9,5.4,1.8/                       0023 0027
      DATA PS/1.022,2.,4.,8.,14.,25.,43.,69.,99.,128.,151.,165.,176.,182 0024 0028
     1.,185.,186.5,187.25,187.46875,187.506,187.6385/                   0025 0029
      DATA TS/24.845,27.07,29.81,33.07,36.18,39.96,44.12,48.33,51.97,54. 0026 0030
     179,56.72,57.80,58.57,58.99,59.18,59.29,59.34,59.353,59.356,59.4/  0027 0031
      DATA TL/24.846,27.175,29.310,31.299,33.176,34.962,36.672,38.317,  0028 0032
     139.904,41.456/                                                    0029 0033
      DATA HS/-157.82,-136.56,-115.76,-95.46,-75.59,-56.08,-36.93,-17.99 0030 0034
     1,0.69,19.30/                                                      0031 0035
      DATAAA/10720.,10750.,10780.,12690.,12720.,12750.,14780.,14780.,148 0032 0036
     110.,17060.,16960.,16960.,19700.,19330.,19260.,10720.,10720.,10730. 0033 0037
     2,12710.,12700.,12700.,14920.,14810.,14780.,17620.,17140.,17030.,21 0034 0038
     3390.,19940.,19580.,10720.,10720.,10720.,12730.,12710.,12700.,15020 0035 0039
     4.,14910.,14850.,18000.,17550.,17330.,22560.,21170.,20520.,10720.,1 0036 0040
     50720.,10720.,10720.,11520.,11500.,11500.,11520.,12360.,12320.,1291 0037 0041
     60.,12320.,13260.,13180.,13150.,13160.,14250.,14090.,14040.,14050., 0038 0042
     715400.,15090.,14980.,14940.,16780.,16220.,16030.,16030.,18500.,175 0039 0043
     840.,17200.,17120.,20680.,19110.,18560.,18430.,23480.,21030.,20170. 0040 0044
     9,19980.,27100.,23410.,22100.,21520.,10720.,10720.,10720.,10720./  0041 0045
      DATAAB/10720.,11540.,11520.,11520.,11510.,11510.,12410.,12370.,123 0042 0046
     160.,12350.,12340.,13390.,13300.,13260.,13230.,13220.,14520.,14330. 0043 0047
     2,14250.,14210.,14170.,15900.,15560.,15400.,15310.,15250.,17680.,17 0044 0048
     3060.,16780.,16620.,16500.,20030.,18970.,18490.,18210.,18020.,23180 0045 0049
     4.,21440.,20670.,20210.,19890.,27390.,24680.,23470.,22750.,22260.,3 0046 0050
     52960.,28900.,27080.,26000.,25260.,1663.,1748.,3407.,3524.,5151.,52 0047 0051
     672.,6935.,7056.,8783.,8901.,10720.,10820.,419.7,385.5,377.1,385.1, 0048 0052
     7401.5,830.7,823.8,826.2,835.9,851.,1261.,1270.,1282.,1297.,1315.,1 0049 0053
     8668.,1687.,1706.,1724.,1743.,102.2,129.1,159.,159.9,182.2,209.4,22 0050 0054
     91.1,238.9,263.6,284.8,299.1,321.3,350.8,362.7,383.,419.,429.4/    0051 0055
      DATAAC/446.8,88.34,86.3,87.94,91.61,96.51,102.2,162.4,155.2,152.9, 0052 0056
     1153.5,156.1,159.9,232.5,224.,219.6,218.2,218.9,221.1,299.,291.1,28 0053 0057
```

********   HYENTH   ********

```
26.2,284.,283.7,284.8,364.3,357.6,393.2,350.8,350.1,350.8,491.8,424   0054 0058
3.8,422.9,420.9,418.9,419.,126.4,109.2,98.33,91.88,88.34,201.9,188.   0055 0059
44,177.2,168.7,162.4,265.3,255.1,246.2,238.6,232.5,325.1,317.4,310.   0056 0060
54,304.2,299.,384.9,378.8,373.4,368.5,364.3,448.8,444.5,440.3,436.,   0057 0061
6431.8,222.2,174.7,126.4,266.9,233.7,201.9,313.5,288.4,265.3,362.4,   0058 0062
7342.7,325.1,414.5,398.7,384.9,465.8,457.3,448.8,-36.57,-.635,35.58   0059 0063
8,-24.27,11.37,46.52,-10.41,24.51,59.14,5.035,39.1,73.14,22.05,54.9   0060 0064
99,88.32,40.42,72.09,104.7,59.98,90.22,122.,80.6,109.3,140.1,102.2/   0061 0065
 DATAAD/129.1,159.,-85.55,-76.17,-66.88,-57.67,-48.54,-39.65,-76.14   0062 0066
1,-67.02,-57.86,-48.74,-39.63,-30.42,-65.46,-56.63,-47.73,-38.82,-2   0063 0067
29.88,-20.81,-53.51,-45.09,-36.51,-27.86,-19.15,-10.41,-40.29,-32.3   0064 0068
38,-24.23,-15.89,-7.47,1.17,-25.74,-18.56,-10.9,-2.96,5.14,13.54,-9   0065 0069
4.88,-3.58,3.41,10.88,18.6,26.64,7.34,12.46,18.68,25.57,32.85,40.42   0066 0070
5,25.89,29.54,34.83,41.04,47.82,54.09,45.69,47.61,51.79,57.23,63.43   0067 0071
6,70.29,66.61,66.57,69.53,74.12,79.68,86.,88.34,86.3,87.94,91.61,96   0068 0072
7.51,102.2,-129.4,-124.6,-119.8,-115.,-110.3,-105.5,-100.8,-96.08,-   0069 0073
891.37,-119.4,-114.9,-110.4,-105.8,-101.2,-96.51,-91.88,-87.21,-82.   0070 0074
941,-107.4,-103.4,-99.26,-94.98,-90.6,-86.19,-81.71,-77.23,-72.58/    0071 0075
 DATAAE/-92.77,-89.71,-86.23,-82.46,-78.47,-74.37,-70.19,-65.93,-61   0072 0076
1.48,-73.53,-72.8,-70.77,-67.93,-64.6,-61.,-57.2,-53.28,-49.11,-41.   0073 0077
228,-49.57,-51.87,-50.91,-48.74,-45.9,-42.7,-39.24,-35.44,39.,-18.9   0074 0078
3,-26.49,-30.41,-30.39,-28.88,-26.55,-23.74,-20.45,46.52,40.72,37.0   0075 0079
41,34.66,33.21,32.38,32.04,32.06,32.49,88.56,79.51,72.99,68.32,65.0   0076 0080
51,62.69,61.11,60.1,59.63,126.4,117.8,109.2,103.8,98.33,95.1,91.88,   0077 0081
690.11,88.34,-30.39,-29.64,-28.88,-27.72,-26.55,-25.15,-23.74,-22.1   0078 0082
77,-20.45,-23.63,-23.33,-22.72,-21.84,-20.77,-19.57,-18.24,-16.82,-   0079 0083
815.17,-16.47,-16.59,-16.27,-15.65,-14.8,-13.76,-12.58,-11.27,-9.88   0080 0084
9,-8.89,-9.51,-9.56,-9.23,-8.6,-7.75,-6.72,-5.57,-4.14,-.85,-2.07/    0081 0085
 DATAAF/-2.56,-2.55,-2.18,-1.55,-.7,.3,1.6,7.68,5.74,4.76,4.38,4.45   0082 0086
1,4.85,5.5,6.34,7.34,16.7,13.93,12.37,11.57,11.3,11.45,11.88,12.53,   0083 0087
213.52,26.21,22.5,20.28,19.02,18.39,18.24,18.43,18.89,19.7,36.36,31   0084 0088
3.44,28.52,26.72,25.7,25.22,25.15,25.4,25.89,46.52,40.72,37.01,34.6   0085 0089
46,32.21,32.38,32.04,32.06,32.49,95.17,80.15,65.54,53.23,44.02,37.3   0086 0090
58,32.55,28.95,26.21,115.7,104.1,92.37,81.05,70.96,62.55,55.86,50.6   0087 0091
62,46.52,132.6,123.2,113.6,104.,94.83,86.45,79.09,72.87,67.54,147.7   0088 0092
7,139.7,131.4,123.3,115.3,107.6,100.5,94.17,88.56,161.6,154.5,147.3   0089 0093
8,140.2,133.2,126.3,119.8,113.7,107.5,174.7,168.7,162.6,156.6,150.5   0090 0094
9,144.5,138.5,132.4,126.4,177.4,161.6,143.4,121.8,95.2,186.3,171.7/   0091 0095
 DATAAG/155.4,136.8,115.7,195.1,181.6,166.8,150.5,132.6,204.,191.4,   0092 0096
1177.9,163.3,147.7,212.9,201.2,188.7,175.5,161.6,222.2,210.3,198.5,   0093 0097
2186.6,174.7,-30.41,-30.63,-30.67,-30.59,-30.39,-22.5,-23.09,-23.44   0094 0098
3,-23.6,-23.63,-13.86,-14.96,-15.7,-16.18,-16.47,-4.37,-6.14,-7.4,-   0095 0099
48.28,-8.89,6.14,3.46,1.54,.16,-.85,17.79,13.93,11.18,9.17,7.68,30.   0096 0100
552,25.28,21.52,18.77,16.7,44.02,37.38,32.55,28.95,26.21,-26.48,-27   0097 0101
6.26,-27.92,-28.45,-28.92,-29.3,-29.62,-29.88,-30.1,-30.28,-30.41,-   0098 0102
721.02,-22.11,-23.,-23.75,-24.38,-24.9,-25.36,-25.73,-26.06,-26.32,   0099 0103
8-26.46,-14.94,-16.45,-17.67,-18.68,-19.53,-20.25,-20.85,-21.37,-21   0100 0104
9.81,-22.19,-22.5,-8.03,-10.16,-11.84,-13.21,-14.34,-15.28,-16.09/    0101 0105
 DATAAH/-16.77,-17.35,-17.85,-18.18,.01,-3.05,-5.37,-7.22,-8.72,-9.   0102 0106
196,-11.01,-11.9,-12.66,-13.31,-13.86,9.41,5.12,1.87,-.61,-2.6,-4.2   0103 0107
23,-5.58,-6.72,-7.69,-8.52,-9.12,20.71,14.43,10.1,6.72,4.1,1.99,.26   0104 0108
3,-1.19,-2.42,-3.47,-4.37,33.45,25.24,19.25,14.8,11.45,8.74,6.54,4.   0105 0109
471,3.17,1.87,.88,45.95,36.79,29.44,23.8,19.44,16.07,13.29,11.02,9.   0106 0110
512,7.51,6.14,56.31,47.94,39.94,33.35,28.09,23.89,20.51,17.76,15.44   0107 0111
6,13.49,11.96,66.66,58.21,50.23,43.14,37.17,32.25,28.21,24.89,22.11   0108 0112
7,19.78,17.79,74.56,67.32,59.73,52.62,46.26,40.81,36.2,32.35,29.11,   0109 0113
826.37,24.15,82.45,75.41,68.34,61.5,55.11,49.35,44.34,40.03,36.35,3   0110 0114
93.21,30.52,88.81,82.65,76.12,69.66,63.44,57.66,52.4,47.78,43.74/     0111 0115
```

********* HYENTH *********

```
      DATAAI/40.24,37.27,95.17,89.16,83.15,77.23,71.98,65.54,60.61,55.69
     1,51.39,47.7,44.02,104.9,98.01,90.23,81.37,71.12,59.16,45.59,31.88,
     220.95,113.3,107.3,100.8,93.63,85.67,76.79,66.95,56.39,45.95,120.4,
     3115.8,110.2,104.2,97.75,90.73,83.18,75.1,66.66,127.6,123.6,118.7,1
     413.4,107.8,102.1,95.85,89.29,82.45,134.8,130.4,126.1,121.8,116.4,1
     511.1,105.8,100.5,95.17,65.16,51.82,-3.47,-10.85,-14.26,-16.57,-18.
     637,-19.82,-21.02,-22.04,-22.92,-23.69,-24.38,-24.99,-25.54,-26.04,
     7-26.48,74.13,67.69,57.36,39.09,5.99,-3.23,-7.63,-10.52,-12.65,-14.
     837,-15.78,-16.98,-18.,-18.9,-19.69,-20.39,-21.02,83.09,77.17,70.3,
     961.77,50.17,31.98,12.48,3.55,-1.09,-4.43,-6.93,-8.9,-10.49,-11.86/
      DATAAJ/-13.03,-14.04,-14.94,89.04,84.56,79.06,72.84,65.52,56.54,44
     1.99,30.6,17.92,9.94,4.89,1.51,-1.24,-3.43,-5.22,-6.72,-8.03,94.98,
     290.77,86.16,81.07,75.42,69.01,61.59,52.84,42.58,31.64,22.32,15.46,
     310.57,6.94,4.28,1.95,.01,99.84,96.27,92.24,87.87,83.15,77.97,72.25
     4,65.86,58.68,50.67,42.06,33.59,26.31,20.44,15.88,12.29,9.41,104.7,
     5101.3,97.67,93.82,89.72,85.3,80.54,75.38,69.76,63.63,56.98,49.93,4
     62.76,35.94,29.96,24.89,20.71,109.1,105.9,102.6,99.07,95.49,91.5,87
     7.5,82.95,78.39,73.16,67.98,62.13,56.27,50.2,44.14,38.79,33.45,71.,
     837.3,0.,0.,0.,84.3,57.3,0.,0.,0.,97.6,72.9,43.,0.,0.,110.9,89.8,62
     9.,0.,0.,124.2,107.4,83.28,48.,0.,137.5,123.5,105.9,81.,0.,150.7/
      DATAAK/138.8,124.9,108.5,85.9,164.1,153.6,142.,128.9,113.3,177.4,1
     167.9,158.,147.,134.8,78.8,62.3,45.4,0.,0.,89.8,78.3,61.,41.9,29.1,
     298.8,90.1,78.9,60.1,27.8,106.7,99.5,91.1,80.6,65.2,114.,107.8,100.
     38,92.8,83.1,120.8,115.2,109.2,102.5,95.,19700.,19330.,19260.,23000
     4.,22040.,21810.,27350.,25260.,24730.,21390.,19940.,19580.,27180.,2
     53580.,22680.,36230.,28590.,26650.,22560.,21170.,20520.,30320.,2664
     60.,25020.,42210.,35060.,31640.,32960.,28900.,27090.,26000.,25260.,
     740130.,34330.,31710.,30130.,29050.,49050.,41170.,37530.,35320.,338
     810.,59610.,49530.,44710.,41750.,39690.,71190.,59240.,53210.,49420.
     9,46750.,82920.,69950.,62910.,58310.,55020.,27100.,23410.,22100./
      DATAAL/21520.,31750.,26350.,24440.,23590.,37610.,30000.,27280.,260
     170.,44900.,34480.,30730.,29050.,53610.,39900.,34870.,32610.,63710.
     2,46340.,39810.,36850./
C
      P=PRES
      IF(P.LT.1.0) P=1.0
      IF(P.LT.1.0) P=1.0
      T=TEMP
      IF(T.LT.90.0) GO TO 10
      IF(T.LT.3000.0) GO TO 5
      IF(T.GE.6000.0) T=5999.99999
      IF(P.LT.30.0) GO TO 3
      IF(P.LT.80.0) GO TO 2
      IF(P.LT.500.0) GO TO 1
      N=1
      NI=25
      GO TO 33
    1 N=2
      NI=26
      GO TO 33
    2 N=3
      NI=27
      GO TO 33
    3 IF(P.LT.5.0) GO TO 4
      N=4
      NI=29
      GO TO 33
    4 N=5
```

********* HYENTH *********

```
      NI=28
      GO TO 33
    5 IF(T.LT.180.0) GO TO 70
      IF(T.LT.500.0) GO TO 6
      N=6
      GO TO 33
    6 N=7
      GO TO 33
   70 IF(P.LT.1175.68) GO TO 8
      IF(P.LT.2645.28) GO TO 7
      N=8
      GO TO 33
    7 N=9
      GO TO 33
    8 IF(P.LT.587.84) GO TO 9
      N=10
      GO TO 33
    9 N=11
      GO TO 33
   10 IF(P.LT.1175.68) GO TO 12
      IF(P.LT.2645.28) GO TO 11
      N=12
      GO TO 30
   11 N=13
      GO TO 30
   12 IF(T.GE.59.4) GO TO 15
      N=14
      IF(P.GE.187.6385) GO TO 30
      DO 13 I=2,20
      IF(P-PS(I))14,14,13
   13 CONTINUE
      I=20
   14 TM=TS(I-1)+(TS(I)-TS(I-1))*(P-PS(I-1))/(PS(I)-PS(I-1))
      IF(T.LE.TM) GO TO 30
      GO TO 23
   15 IF(P.LT.293.92) GO TO 20
      IF(P.GE.587.84) GO TO 18
      IF(T.GE.72.0) GO TO 17
      IF(P.LT.440.88) GO TO 16
      N=19
      GO TO 33
   16 N=20
      GO TO 33
   17 N=17
      GO TO 33
   18 IF(T.LT.75.6) GO TO 19
      N=15
      GO TO 33
   19 N=16
      GO TO 33
   20 IF(T.GE.72.0)GO TO 22
      IF(P.LT.176.352) GO TO 23
      IF(T.LT.65.7) GO TO 21
      N=21
      GO TO 33
   21 N=22
      GO TO 33
   22 N=18
```

********   HYENTH   ********

```
      GO TO 33
   23 N=23
      IF(P.GE.117.568.AND.T.LE.63.) N=24
      GO TO 33
   30 F=P/587.84
      I=F
      IF(I.GT.8) I=8
      FI=I
      F=F-FI
      FP=1.0-F
      TQ=FP*TL(I+1)+F*TL(I+2)
      IF(T.GE.TQ) GO TO 33
      HYENTH=FP*HS(I+1)+F*HS(I+2)
      RETURN
   33 IF(T.LE.5000.)N1=N
      FP=(P-BP(N))/DP(N)
      IP=FP
      IF(IP.GT.MX(N)) IP=MX(N)
      FI=IP
      F=FP-FI
      FP=1.0-F
      FT=(T-BT(N1))/DT(N)
      IT=FT
      FI=IT
      FF=FT-FI
      FT=1.0-FF
      I=IT*JP(N)+IP+LOC(N1)
      J=I+JP(N)
      HYENTH=FP*FT*H(I)+F*FT*H(I+1)+FP*FF*H(J)+F*FF*H(J+1)
      RETURN
      END
```

```
C                   *****************************************
C                   * ROUTINE NAME - TABLE INPUT ROUTINE             *
C                   * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                   * PPOGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                   * DATE CODED   - 3/4/70                       *
C                   * REVISED      - JUNE  1971 .. DEC. 1971      *
C                   * PROGRAMMER   - J. MCKAY  19-43  201  45178 *
C                   *****************************************
C
      SUBROUTINE INTAB
C
      INTEGER TITLE,OFT
C
      LOGICAL JP,PAGE
C
      INCLUDE CIOUNT
      INCLUDE CTAB
      INCLUDE CTABA
C
      DIMENSION TITLE(4),ICMT(14),LABV(3,5),LABX(3),LABY(3),II(5)
C
C         ***** INPUT FORMATS
C
 5000 FORMAT (4A6,2I6,6X,I6)
 5010 FORMAT(13A6,A2)
 5020 FORMAT(3A6,I7,5E10.0)
 5030 FORMAT(6A6,2E12.0)
 5040 FORMAT(3I6)
 5050 FORMAT(6E12,0)
 5060 FORMAT (5I5)
C
C         ***** OUTPUT FORMATS
C
 6000 FORMAT('0',T43,'****** E C H O   O F   T A B L E   D A T A ******',
     1 // T54,4A6,T100,'TABLE NO.'I4//)
 6010 FORMAT('0 **ERROR** THE NUMBER OF DIMENSION IS WRONG, ND = ',I5)
 6020 FORMAT(T26,13A6,A2)
 6030 FORMAT('0 **ERROR** THE NUMBER OF POINT IS WRONG, NP = ',I5)
 6040 FORMAT (T36,3A6,2X,5F13.4)
 6050 FORMAT('0 **ERROR** THE NUMBER OF DATA POINTS IS WRONG, NV = ',I5)
 6060 FORMAT('0 **ERROR** THE TABLE TYPE IS WRONG. TYPE = ',I5)
 6070 FORMAT('0 **DIAGNOSTIC** THE NUMBER OF INTERPOLATION POINTS IS WRO
     ING, NIP = ',I5,'. NIP IS SET EQUAL TO ',I5)
 6080 FORMAT (T52,3A6,2X,F12.4)
 6090 FORMAT(T52,F12.4,5XF12.4)
 6100 FORMAT(T52,F12.4)
 6110 FORMAT(T38,F12.4,2XF12.4,5XF12.4,2XF12.4)
 6120 FORMAT(T38,F12.4,2XF12.4)
 6130 FORMAT('0'T53,'COEFICIENTS',6X,'COEFICIENTS'/' ')
 6140 FORMAT('0'T44,'X',13X,'Y',16X,'X',13X,'Y')
 6150 FORMAT('0 **ERROR** THE ABOVE TABLE NUMBER IS LESS THAN 0 OR GREAT
     1ER THAN 50'//)
 6160 FORMAT('0 **DIAGNOSTIC** THE ABOVE TABLE HAS ALREADY BEEN INPUT. T
     1HIS TABLE SHALL REPLACE THE PREVOUS TABLE.')
 6170 FORMAT('0 *ERROR* THE TOTAL SIZE OF THE TABLES HAS EXCEEDED  7000.
     1 THE REQUIRED SIZE IS ',I6,'. RUN TERMINATED.')
 6180 FORMAT ('0'75X'TOTAL TABLE STORAGE ='I6)
```

```
 6200 FORMAT ('0'56X'TABLE INPUT SUMMARY'// 28X'TABLE'6X'TITLE OF TABLE'
     1 12X'NUMBER OF' 6X'NUMBER OF'6X'NUMBER OF'/ 27X'NUMBER'T66,
     2 'DIMENSIONS'5X'SUBTABLES'9X'WORDS'/)
 6210 FORMAT (28XI4,2X,4A6,3I15)
 6220 FORMAT ('0' T90,'NO. WORDS THIS TABLE' I4)
C
C         ***** INITIALIZE THE ROUTINE
C
C                             NPRT = 0 PRINT ALL TABLES (1 TABLE/PAGE)
C                             NPRT = 1 PRINT NO TABLE OUTPUT
C                             NPRT > 1 PRINT ALL TABLES (NO PAGE EJECT)
C                             NPRT2 > 0 AND NPRT = 1 PRINT SUMMARY
C
      INTAP = NTAPE1
      IOTAP = NTAPE1
      READ (IIN,5060) IFT,OFT,NPRT,NPRT2
      IF (IFT .EQ. 0) GO TO 5
      IF (IFT .GT. 2) INTAP = IFT
      READ (INTAP) TLA,NV
      READ (INTAP) (TABLE(JKM),JKM=1,NV)
      REWIND INTAP
      KMURD1 = NV
      GO TO 350
C
    5 KMURD1 = 0
      IF (NPRT .NE. 1) GO TO 2
      IF (NPRT2 .EQ. 0) GO TO 2
      IF (PAGE(0)) WRITE (IOT,6200)
    2 CONTINUE
C         ***** START OF TABLE INPUT LOOP
C
      DO 290 II=1,NTBN
      KMUD = KMURD1
C
C         ***** INPUT TITLE - THE TABLE TITLE.
C                   *       ND    - THE NUMBER OF DIMENSIONS OF THIS TABLE.
C                   *       NC    - THE NUMBER OF COMMENT CARDS FOR THIS TABLE
C                   *       IP    - PLOT OPTION, NOT IN MATH MODEL
C         *****                          (SEE AUXILLARY TABLE PROGRAM)
C
      READ (IIN,5000) TITLE,ND,NC,NT
      IF(ND.LE.0) GO TO 300
      IF (NPRT .EQ. 1) GO TO 1001
      IF (PAGE(0)) WRITE (IOT,6000) TITLE,NT
      JP = PAGE (5)
 1001 CONTINUE
      KMURD1 = KMURD1 + 1
      IF (NT .GT. 0 .AND. NT .LE. NTBN) GO TO 4
      WRITE(6,6150)
      CALL EXIT
    4 IF(TLA(NT).LE.0) GO TO 6
      WRITE(6,6160)
    6 TLA(NT) = KMURD1
      ITABLE(KMURD1) = ND
C
C         ***** TEST INPUT VALUE.
C
      IF (ND .GT. 1 .AND. ND .LT. 7) GO TO 20
```

```
      WRITE(6,6010) ND
      CALL EXIT
C
C          ***** IF THERE ARE ANY COMMENT CARDS FOR THIS PLOT PRINT
C          ***** THEM OUT.
C
   20 IF (NC .LE. 0) GO TO 40
C
      DO 30 I2 = 1,NC
      READ(5,5010) ICMT
      IF (NPRT .EQ. 1) GO TO 30
      IF (PAGE(1)) WRITE (IOT,6000) TITLE,NT
      WRITE(6,6020) ICMT
   30 CONTINUE
C
C          ***** INITIALIZE THE INPUT OF THN ND - 2 INDEPENDENT
C          ***** VARIABLES.
C
   40 NDM2 = ND - 2
      NXYT = 1
      IF(NDM2.EQ.0) GO TO 90
      IF (NPRT .EQ. 1) GO TO 50
      IF (PAGE(NDM2+1)) WRITE (IOT,6000) TITLE,NT
C
C          ***** INPUT ND - 2 INDEPENDEND VARIABLES.
C
   50 DO 80 I2 = 1,NDM2
      READ(5,5020) (LABV(I,I2),I=1,3),NP,(TAB(I+1,I2),I=1,NP)
      IF(NP.GT.1.AND.NP.LT.6) GO TO 60
      WRITE(6,6030) NP
      CALL EXIT
C
   60 NXYT = NXYT*NP
      IF (NPRT .EQ. 1) GO TO 70
      WRITE(6,6040) (LABV(I,I2),I=1,3),(TAB(I+1,I2),I=1,NP)
   70 CONTINUE
      KMURD1 = KMURD1 + 1
      ITABLE(KMURD1) = NP
      DO 75 I3=1,NP
      KMURD1 = KMURD1 + 1
C
C          ***  STORE THE (ND-2) INDEPENDENT VARIABLES IN TABLE ARRAY
C
   75 TABLE(KMURD1) = TAB(I3+1,I2)
   80 CONTINUE
C
C          ***** IF PLOTING IS DESIRED, INPUT THE PLOT LABELS AND
C          ***** THE MIN/MAX VALUES OF THE INDEPENDENT VARIABLE.
C
   90 READ(5,5030) LABX,LABY,XMIN,XMAX
C
C          ***** START OF THE LOOP TO INPUT THE VALUES FOR EACH
C                * OF THE TWO DIMENSIONAL TABLES WHICH MAKE UP THE MAIN
C          ***** ND DIMENSIONAL TABLES.
C
      II(1) = -1
      DO 100 I2 = 2,5
  100 II(I2) = - 1
```

```
C
      DO 280 I2 = 1,NXYT
C
C         ***** INPUT THE 2D TABLE HEADER CARD AND VERIFY THE DATA.
C               * NV   - NUMBER OF DATA POINTS IN THE TABLE
C               * TYPE - TABLE TYPE (SEE CTAB)
C         ***** NIP  - NUMBER OF POINTS TC BE USED FOR INTERPOLATION.
C
      READ(5,5040) NV,TYPE,NIP
      IF (NPRT .NE. 0) GO TO 1004
      IF (PAGE(0)) WRITE (IOT,6000) TITLE,NT
 1004 CONTINUE
C
      IF (NV .GT. 0 .AND. NV .LE. NSBZ) GO TO 110
      WRITE(6,6050) NV
      CALL EXIT
C
  110 IF(TYPE.EQ.0.OR.TYPE.EQ.1) GO TO 120
      WRITE(6,6060) TYPE
      CALL EXIT
C
  120 IF(TYPE.EQ.0) GO TO 130
      IF(NIP.GT.1.AND.NIP.LE.NV) GO TO 130
      WRITE(6,6070) NIP,NV
      NIP = NV
C
C         ***** OUTPUT THE HEADER DATA TO DRUM
C
  130 ITABLE(KMURD1+1) = NV
      ITABLE(KMURD1 + 2) = TYPE
      ITABLE(KMURD1+3) = NIP
      KMURD1 = KMURD1 + 3
C
C         ***** INPUT, ECHO AND OUTPUT TO DRUM THE 2D TABLES.
C
      IF (NPRT .EQ. 1) GO TO 170
      II(NDM2) = MOD(II(NDM2) + 1,ITAB(1,NDM2))
      IF(NDM2.EQ.0) GO TO 170
      IF(NDM2.EQ.1) GO TO 150
      NDM3 = NDM2 - 1
C
      DO 140 I3 = NDM3,1,-1
      IF(II(I3+1).GT.0) GO TO 150
      II(I3) = MOD(II(I3) + 1,ITAB(1,I3))
  140 CONTINUE
C
  150 DO 160 I3 = 1,NDM2
      IDX = II(I3) + 2
      WRITE(6,6080) (LABV(I,I3),I=1,3),TAB(IDX,I3)
  160 CONTINUE
C
  170 IF(TYPE.EQ.1) GO TO 210
C
C         ***** COEFICIENT TABLE INPUT
C
      READ(5,5050) (XTAB(I),I=1,NV)
      DO 175 I3 = 1,NV
      KMURD1 = KMURD1 + 1
```

```
          TABLE(KMURD1) = XTAB(I3)
  175 CONTINUE
      IF (NPRT .EQ. 1) GO TO 280
      WRITE(6,6130)
C
      IF(NV.GT.50) GO TO 180
      N1 = 1
      N2 = NV
      GO TO 190
C
  180 N2 = NV - 50
      WRITE(6,6090) (XTAB(I),XTAB(I+50),I=1,N2)
      N1 = N2 + 1
      IF (N1 .GT. 50) GO TO 280
      N2 = 50
  190 WRITE(6,6100) (XTAB(I),I=N1,N2)
C
C           ***** DESCRETE TABLE INPUT
C
  210 READ(5,5050) (XTAB(I),YTAB(I),I=1,NV)
      DO 215 I3 = 1,NV
      KMURD1 = KMURD1 + 1
      TABLE(KMURD1) = XTAB(I3)
      TABLE(KMURD1+NV) = YTAB(I3)
  215 CONTINUE
      KMURD1 = KMURD1 + NV
      IF (NPRT .EQ. 1) GO TO 1005
      WRITE(6,6140)
 1005 CONTINUE
      IF(NV.GT.50) GO TO 220
      N1 = 1
      N2 = NV
      GO TO 230
C
  220 N2 = NV - 50
      WRITE(6,6110) (XTAB(I),YTAB(I),XTAB(I+50),YTAB(I+50),I=1,N2)
      N1 = N2 + 1
      IF (N1 .GT. 50) GO TO 280
      N2 = 50
  230 CONTINUE
      IF (NPRT .EQ. 1) GO TO 280
      WRITE (6,6120) (XTAB(I),YTAB(I),I=N1,N2)
C
  280 CONTINUE
      KWRD = KMURD1 - KMUD
      IF (NPRT .EQ. 1 .AND. NPRT2 .NE. 0) WRITE (IOT,6210) NT,TITLE,
     1                                                     ND,NXYT,KWRD
      IF (NPRT .NE. 1) WRITE (IOT,6220) KWRD
C
  290 CONTINUE
C
  300 IF (KMURD1 .LE. MXWRD) GO TO 310
      WRITE(6,6170) KMURD1
      CALL EXIT
  310 IF (OFT .EQ. 0) GO TO 350
      IF (OFT .GT. 2) IOTAP = OFT
      WRITE (IOTAP) TLA,NV
      WRITE (IOTAP) (TABLE(JKM),JKM=1,KMURD1)
```

```
*********    INTAB    *********

          END FILE IOTAP
          REWIND IOTAP
      350 WRITE (IOT,6180) KMURDI
          RETURN
C
          END
```

********* SUBRØUTINE LIQRES

```
C                  ************************************************
C                  * ROUTINE NAME - LIQUID RESIDUALS              *
C                  *                DETERMINATION ROUTINE         *
C                  * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                  * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                  * DATE CODED   - 5/15/70                       *
C                  * REVISED      - JULY  1972                    *
C                  * PROGRAMMER   - J. MCKAY  D1943  201  45178 *
C                  ************************************************
C
      SUBROUTINE LIQRES
C
      INCLUDE CTANK
C
      DIMENSION CST(2,3)
C
      DATA (CST(I,1),I=1,2) / .022   ,.0035    /
      DATA (CST(I,2),I=1,2) / .001   ,.000858  /
      DATA (CST(I,3),I=1,2) / .00075,.0001142 /
C
C         ***** THREE TYPES OF ACQUISITION DEVICES ARE AVAILABLE
C
C                    IDX = 1    SURFACE TENSION DEVICE
C                    IDX = 2    POSITIVE DISPLACEMENT DEVICE
C                    IDX = 3    DIELECTROPHORETIC DEVICE
C
      DO 1000 I1=1,2
      IDX = SATYPE(I1)
      IF (I1 .EQ. 2 .AND. IDX .GT. 1) GO TO 100
C                   OXY, SURFACE TENSION ONLY
      IF (WTOTP .LE. 100000.) GO TO 100
      WLR = 2200. + .008*(WTOTP - 100000.)
      GO TO 200
  100 WLR(I1) = CST(I1,IDX)*WTOTP(I1)
  200 IF (WLRT(I1,1) .GT. WLR(I1)) WLR(I1) = WLRT(I1,1)
 1000 CONTINUE
      RETURN
      END
```

******** SUBRØUTINE LØCAT

```
C                   ** * * * * * * * * * * * ** * * * * * * * * * * *
C                   * ROUTINE NAME - LOCATE AND INPUT THE TABLE    *
C                   *                TO BE LOOKED-UP                *
C                   * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                   * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                   * DATE CODED   - 3/9/70                        *
C                   ** * * * * * * * * * * * ** * * * * * * * * * * *
C
      SUBROUTINE LOCAT (MNT)
C
C         ***** EXPLANATION OF THE CALLING SEQUENCE
C             *
C             * NT - THE TABLE NUMBER WITHIN A MASTER TABLE, THAT IS,
C             *      THE NUMBER OF WORDS TO BE SKIPED TO FIND THE
C         *****      DESIRED TABLE IS 200*(NT-1).
C
      LOGICAL DIAG
C
      INCLUDE CIOUNT
      INCLUDE CTAB
      INCLUDE CTABA
C
      NT = IABS (MNT)
      IF (DIAG(0,6HLOCATE)) WRITE (IOT,6000) NT,MNT,IDX1
      IDX = IDX1
      NTM1 = NT - 1
      IF(NTM1.LE.0) GO TO 20
C
      DO 10 I1 = 1,NTM1
      MM = 1
      IF(ITABLE(IDX+1).EQ.1) MM = 2
   10 IDX = IDX + MM*ITABLE(IDX) + 3
C
   20 NV = ITABLE(IDX)
      TYPE = ITABLE(IDX+1)
      NIP = ITABLE(IDX+2)
      IDX = IDX + 2
C
      DO 30 I1 = 1,NV
      IDX = IDX + 1
      IF (MNT .LT. 0) GO TO 26
      XTAB(I1) = TABLE(IDX)
      IF(TYPE.EQ.1) YTAB(I1) = TABLE(IDX+NV)
      GO TO 30
C                             SWITCH DEPENDENT AND INDEP. VARIABLES
   26 XTAB(I1) = TABLE(IDX+NV)
      YTAB(I1) = TABLE(IDX)
   30 CONTINUE
      IF (DIAG(1,6HLOCATE)) WRITE (IOT,6000) NV,TYPE,NIP,IDX,XTAB(1),
     1                                       XTAB(NV),YTAB(1),YTAB(NV)
      RETURN
 6000 FORMAT ('+'14X,4I5,6E15.6)
      END
```

******** SUBRØUTINE LPRØP

```
      SUBROUTINE LPROP(T,P,D,K,H,S,U,Z)
      COMMON /REFPR/ RF(10)
      COMMON /METH/ M
C
C.... ROUTINE TO CALCULATE THE PROPERTIES OF THE LIQUID
C
C     K =1   INPUT IS T + D
C     K =2   INPUT IS T + P
C     K =3   INPUT IS T, P, + D
C
      IF(M.EQ.2)GO TO 1
      CALL VPROP(T,P,D,K,H,S,U,Z)
      RETURN
    1 R =RF(5)
      AK=RF(6)
      IF(K.EQ.1)CALL PFND(T,D,P)
       IF(K.EQ.2)CALL DFND(T,P,D,Z1,0)
      VP=VPN(T)
       CALL DFND(T,VP,DSV,Z1,2)
      CALL VPROP(T,VPP,DSV,1,HSV,SSV,USV,ZSV)
      VSV = 1.0/DSV
       CALL DFND(T,VP,DSL,Z1,1)
      VSL = 1.0/DSL
      CALL DPDTVP(T,VP,DPDT)
      HSL=HSV-T*DPDT*(VSV-VSL)*AK
      SSL=SSV+(HSL-HSV)/T
      USL=USV+(HSL-HSV)-(VP*(VSL-VSV))*AK
      DLD = ALOG(D)
      DLS = ALOG(DSL)
      F1D=R*DLD-FING1(T,D)
      F1S=R*DLS-FING1(T,DSL)
      F2D=FING2(T,D)+R*T*DLD
      F2S=FING2(T,DSL)+R*T*DLS
      S=SSL-(F1D-F1S)*AK
      U=USL+((F2D-F2S)-T*(F1D-F1S))*AK
      H=U + (P/D)*AK
      Z = P/(D*R*T)
      RETURN
      END
```

********* SUBRØUTINE LPRØPB

```
      SUBROUTINE LPROPB(TB,PB,DB,K,HB,SB,UB,ZB)
      COMMON /RFPR/RF(10)
      WT=RF(7)
      T  = TB/1.8
      P  = PB * 6.8947572E+3/1.01325E+5
      D  = DB * 453.59237E-3/(WT * 2.8316847E-2)
      CALL LPROP(T,P,D,K,H,S,U,Z)
      PB = P * 1.01325E+5/6.8947572E+3
      DB = D * WT * 2.8316847E-2/453.59237E-3
      HB = H * 453.59237/(1.0543502E+3 * WT)
      UB = U * 453.59237/(1.0543502E+3 * WT)
      SB = S * 453.59237/(1.0543502E+3 * 1.8 * WT)
      ZB = Z
      RETURN
      END
```

********* SUBRØUTINE LSSCMP

```
      SUBROUTINE  LSSCMP
C
C     *** THIS SUBROUTINE PROVIDES ECLSS CONFIGURATION COMPONENT
C       * SIZING AND PRESSURE DROP DATA FOR OXYGEN AND NITROGEN
C     *** FLUIDS.
C
C
      INTEGER GSTATE
C
      LOGICAL PAGE, JP
C
C
      INCLUDE CACCUM
      INCLUDE CCNFIG
      INCLUDE CCNTRL
      INCLUDE CDCYCL
      INCLUDE CECLSS
      INCLUDE CENG
      INCLUDE CHEX
      INCLUDE CHSORC
      INCLUDE CIOUNT
      INCLUDE CNAMES
      INCLUDE CONST
      INCLUDE CTANK
C
C         ***** INITIALIZE THE ROUTINE
C
      IDX = 0
      ISIGN = 1
      JKM = 0
      HIENTH = 0.0
      WGGTOT(1) = 0.0
      WGGTOT(2) = 0.
      HFTOT(1) = 0.0
      HFTOT(2) = 0.0
      QTREQ(1) = 0.0
      QTREQ(2) = 0.0
      C1 = 1152.0/(GRAVTY*PI**2)
      IF(PAGE(0)) WRITE (IOT,6050)
      WRITE (IOT,6020)
      JP = PAGE (3)
C
C         ***** START OF CONFIGURATION PROCESSING LOOP
C
      DO 1000 II=1,ICNF
      IDX = IDX + ISIGN
      MACH(IDX) = 0.0
      MFLG(IDX) = 6H
      CALL GETCON(IDX)
C
C         ***** BRANCH TO THE REQUIRED CONFIGURATION TYPE, SEE CCNFIG
C         ***** FOR BRANCH DEFINITIONS.
C
      GO TO (100,200,300,400,450,500,450,450,400,405,600,700,800,900,
     1 230,250,270,1100), CFUNCT
C
C         ***** SETUP THE GAS TYPE *****
```

```
C
  100 IGAS = CFTYPE
      GSTATE = ICNFIG(5)
      IF (IGAS .EQ. JKM) GO TO 110
      JKM = IGAS
      ISIGN = 1
      ISTRT(IGAS) = IDX + 1
      JX = 0
  110 CONTINUE
C
      IF(IGAS.EQ. 2.AND.GSTATE.EQ.1) GO TO 111
      GO TO 112
  111 IF(PAGE(0)) WRITE (IOT,6051)
      WRITE (IOT,6020)
      JP = PAGE(3)
C
  112 CONTINUE
C
      IF(I1.EQ.1) GO TO 999
      IF(IGAS.EQ. 2.AND.GSTATE.EQ.1) GO TO 999
      PRES(IDX) = PRES(IDX - ISIGN)
      WDOTN(IDX) = WDOTN(IDX-ISIGN)
      TEMP(IDX) = TEMP(IDX - ISIGN)
      GO TO 999
C
C       ***** PROCESS THE ECLSS *****
C
  270 WDOTN(IDX) = WDOTI(IGAS)
      PRES(IDX)  = PLSNOM(IGAS)
      TEMP(IDX)  = TLSNOM(IGAS)
      GO TO 999
C
C         ***** PROCESS A LINE *****
C
  300 FLD = FRCOEF(IDX)*LOD(IDX)/DIAM(IDX)
      LDV = CFTYPE/10
      CFTYPE = CFTYPE - LDV * 10
  310 WDOTN(IDX) = WDOTN(IDX-ISIGN)
      TEMP(IDX) = TEMP(IDX-ISIGN)
      GO TO 510
C
C         ***** PROCESS A CONTROL *****
C
  400 FLD = FRCOEF(IDX)*LOD(IDX)
      IDV = CFTYPE /10
      CFTYPE = CFTYPE - IDV * 10
      DIAM(IDX) = AMIN1(DIAM(IDX+1),DIAM(IDX-1))
      GO TO 310
C
C         *****  PROCESS A REGULATOR  *****
C
  405 FLD = FRCOEF(IDX)*LOD(IDX)
      IDV = CFTYPE /10
      CFTYPE = CFTYPE - IDV * 10
      DIAM(IDX) = AMIN1(DIAM(IDX+1),DIAM(IDX-1))
      WDOTN(IDX) = WDOTN(IDX-ISIGN)
      TEMP(IDX) = TEMP(IDX-ISIGN)
C
```

```
      IX = IDX - ISIGN
      IF(APRES(IGAS) .EQ. 0.0) GO TO 406
      DLPREG = (APRES(IGAS) - ANDELP(IGAS)/2.0) - PRES(IX)
      PRES(IDX) = PRES(IX) + ISIGN * DLPREG
      GO TO 561
C
  406 CONTINUE
      DLPREG = HEXCOP(1,IGAS) - PRES(IX)
      PRES(IDX) = PRES(IX) + ISIGN * DLPREG
      GO TO 561
C
C          ***** PROCESS A FITTING *****
C
  450 FLD = FRCOEF(IDX) * LOD(IDX)
      LDV = CFTYPE/10
      CFTYPE = CFTYPE - LDV * 10
      DIAM(IDX) = AMIN1(DIAM(IDX+1),DIAM(IDX-1))
      GO TO 310
C
C          ***** PROCESS A TAP *****
C
  500 WDOTN(IDX) = WDOTT(IGAS)
      LDV = CFTYPE/10
      CFTYPE = CFTYPE - LDV * 10
      FLD = FRCOEF(IDX)*LOD(IDX)
      TEMP(IDX) = TEMP(IDX-ISIGN)
      DIAM(IDX) = AMIN1(DIAM(IDX+1),DIAM(IDX-1))
C
C          *****  COMPUTE LINE, CONTROL, FITTING OR TAP DELTA PRESSURE.
C
  510 IX = IDX - ISIGN
C
C          ***** DELTA PRESSURE WHEN GASEOUS
C
C                         CALC. RHO OF GAS
  520 IF(IGAS.EQ.2) GO TO 521
      CALL DENSON(TEMP(IX),PRES(IX),1,RHO,ZEO)
      GO TO 522
  521 CALL DENSON(TEMP(IX),PRES(IX),18,RHO,ZEN)
  522 CONTINUE
C
      DELP = C1*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IDX)**4)
C
C          ***** IF PCT. OF PRESSURE CHANGE EXCEEDS ONE PCT. - RECOMPUTE
C          ***** DELTA-P, IF NOT, COMPUTE THE NEW PRESSURE
C
      IF(DELP/(PRES(IX) + DELP) - 0.01)560,560,530
C
C                         CALC. RHO OF GAS
  530 IF(IGAS.EQ.2) GO TO 531
      CALL DENSON(TEMP(IX),PRES(IX)+DELP/2.0,1,RHO,ZEO)
      GO TO 532
  531 CALL DENSON(TEMP(IX),PRES(IX)+DELP/2.0,18,RHO,ZEN)
  532 CONTINUE
C
      DELP = C1*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IDX)**4)
C
C          ***** AGAIN CHECK PCT. OF PRESSURE CHANGE. IF PCT. EXCEEDS
```

```
C             * 2.8 PCT. COMPUTE THE DELTA-P BY USE OF THE COMPRESSIBLE
C         ***** FLOW EQUATIONS. (REF.-RPL-TDR-64-25.VOL.I.REV.D)
C
      IF(DELP/(PRES(IX) + DELP) - 0.028)560,560,540
C
  540 A = PI*DIAM(IDX)**2/576.0
      IF(IGAS.EQ.2) GO TO 541
      CALL COMFLO(IDX,PRES(IX),TEMP(IX),FLD,A,WDOTN(IX)/CNOPER,1,DELP)
      GO TO 542
  541 CALL COMFLO(IDX,PRES(IX),TEMP(IX),FLD,A,WDOTN(IX)/CNOPER,18,DELP)
  542 CONTINUE
C
      PRES(IDX) = PRES(IX) + ISIGN * DELP
      GO TO 561
C
C         ***** COMPUTE NEW PRESSURE
C
  560 PRES(IDX) = PRES(IX) + ISIGN*DELP
C
C         ***** COMPUTE THE GAS MACH NUMBER
C
C                      CALC. RHO OF GAS
      IF(IGAS.EQ.2) GO TO 551
      CALL DENSON(TEMP(IX),PRES(IDX),1,RHO,ZEO)
      GO TO 552
  551 CALL DENSON(TEMP(IX),PRES(IDX),18,RHO,ZEN)
  552 CONTINUE
C
      IF(IGAS.EQ.2) GO TO 554
      CALL VGVS(IDX,RHO,1)
      GO TO 555
  554 CALL VGVS(IDX,RHO,18)
  555 CONTINUE
C
  561 CONTINUE
C
C         ***** COMPUTE LINE WEIGHT
C
      IF(CFUNCT.EQ.3) GO TO 562
      IF(CFUNCT.EQ.5) GO TO 562
      IF(CFUNCT.EQ.6) GO TO 562
      IF(CFUNCT.EQ.7) GO TO 562
      IF(CFUNCT.EQ.8) GO TO 562
      GO TO 570
C
  562 CALL LWEGHT(IDX,LDV)
C
      GO TO 999
C
C         ***** COMPUTE CONTROL, FITTING OR TAP WEIGHT
C
  570 WEIGHT(IDX) = CFTW (DIAM(IDX),PRES(IDX),IDV)
      GO TO 999
C
C         ***** PROCESS AN ACCUMULATOR *****
C
  600 PRES(IDX) = APRES(IGAS)
      TEMP(IDX) = TEMP(IDX - ISIGN)
```

```
          WDOTN(IDX) = WDOTN(IDX - ISIGN)
          INDXAC(IGAS) = IDX
          GO TO 999
C
C              ***** PROCESS A TANK OR SUPPLY *****
C
      700 CFT = 1
          INDXTK(IGAS) = IDX
C
          IF(SIPRES(IGAS,CFT)) 720,710,720
C
C              ***** IF NO TANK INPUT PRESSURE IS INPUT USE THE VALUE CALC.
C
      710 SIPRES(IGAS,CFT) = PRES(IDX-ISIGN)
          GO TO 740
C
C              ***** CHECK THE TANK INPUT PRESSURE AGANIST THE REQUIRED
C                  * CALCULATED PRESURE. IF THE TANK INPUT PRESSURE IS LESS
C                  * THAN THE CALC. PRESSURE WRITE A DIAGNOSTIC MESSAGE AND
C                  * SET THE TANK INPUT PRESSURE = THE REQUIRED PRESSURE.
C              ***** IF NOT CONTINUE WITH CALCULATIONS.
C
      720 IF(SIPRES(IGAS,CFT) - PRES(IDX-ISIGN)) 730,740,740
C
      730 WRITE (6,6000) SIPRES(IGAS,CFT), PRES(IDX-ISIGN)
          GO TO 710
C
      740 PRES(IDX) = SIPRES(IGAS,CFT)
C
C              ***** DO THE SAME CHECKS FOR THE INPUT TANK TEMPRATURE.
C
          IF(SITEMP(IGAS,CFT)) 760,750,760
      750 SITEMP(IGAS,CFT) = TEMP(IDX-ISIGN)
C
      760 TEMP(IDX) = SITEMP(IGAS,CFT)
C
          WDOTN(IDX) = WDOTN(IDX-ISIGN)
          WEIGHT(IDX) = WTTOT(IGAS)
          GO TO 999
C
      800 GO TO 1000
C
C              ***** PROCESS A HEAT EXCHANGER *****
C
      900 IF(ISIGN.GT.0) GO TO 910
C
          WRITE (IOT,6005) ISIGN
C
      910 CONTINUE
          JX = JX + 1
          JHX = JX
          WDOTN(IDX) = WDOTN(IDX-ISIGN)
          WDOTCF(JX,IGAS) = WDOTN(IDX)
          UCODE(JX,IGAS) = CODE(IDX)
C
          TEMP(IDX) = HEXCIT(JX,IGAS)
          DLPRES = HXCDLP(JX,IGAS)
          PRES(IDX)= PRES(IDX-ISIGN) + DLPRES*ISIGN
```

```
      WEIGHT(IDX) = WHXTOT(JX,IGAS)
C
C        ***** END OF CONFIGURATION PROCESSING LOOP *****
C
  999 CONTINUE
      IF (.NOT. PAGE(1)) GO TO 1998
C
C     *** PAGE HEADER HAS BEEN MOVED TO STATEMENT GROUP 100
C
 1998 CONTINUE
      KFUNCT = FNAME(CFUNCT)
      WRITE(IOT,6030)KFUNCT,CODE(IDX),CFTYPE,CNOPER,CNSTBY,ISIGN,IDX,
     1              IGAS,GSTATE,FRCOEF(IDX),LOD(IDX),DIAM(IDX),
     2              ITHICK(IDX),PRES(IDX),TEMP(IDX),WDOTH(IDX),
     3              WEIGHT(IDX),MACH(IDX),MFLG(IDX)
      IF (PRES(IDX) .GE. 0. .AND. TEMP(IDX) .GE. 0.) GO TO 998
      WRITE (IOT,6040)
      CALL EXIT
  998 CONTINUE
C
  200 CONTINUE
  230 CONTINUE
  250 CONTINUE
 1000 CONTINUE
 1100 CONTINUE
C                   USED BY WEIGHT SUMMARY OUTPUT
      KHEND = IDX - 1
      KOEND = IHSTT - 2
C
C
      RETURN
C
C        ***** OUTPUT FORMATS
C
 6000 FORMAT('0 *DIAGNOSTIC* TANK INPUT PRESSURE IS LESS THAN THE REQUIR
     1ED PRESSURE, TANK PRESSURE SET = REQUIRED PRESSURE,'/15X,'TANK INP
     2UT PRESSURE = ',F7.2,', REQUIRED PRESSURE = ',F7.2)
C
 6005 FORMAT('0 **ERROR** ISIGN =',I3,'THERE IS A CONFIGURATION ERROR'/)
C
 6010 FORMAT('0 *ERROR* A PUMP WAS ENCOUNTERED BUT NO TANK CAN BE FOUND.
     1 PUMP CONFIGURATION INDEX NUMBER = ',I3)
C
 6020 FORMAT('0  F   CODE   FT  NO  NS  IS IDX   G  GS   FCOEF      L/D
     1          DIAM  ITHICK    PRES      TEMP    WDOT  WEIGHT    MACH    M
     2FLAG'/' ')
C
 6030 FORMAT(2XA3,2XA6,I3,6I4,F9.6,F12.4,2F8.4,2F9.2,F8.4,F8.4,F10.7,
     13X,A6)
C
 6040 FORMAT (T44, '***  TERMINATE - NEGATIVE TEMP. OR PRES.  ***')
C
 6050 FORMAT(/T38,'*** SUMMARY OF COMPUTED SYSTEM CONFIGURATION PARAMETE
     1RS ***')
C
 6051 FORMAT(/T32,'*** SUMMARY OF COMPUTED SYSTEM CONFIGURATION PARAMETE
     1RS - CONTD. ***')
C
```

```
*********    LSSCMP    *********
          END
```

******** SUBRØUTINE LWEGHT

```
C                    ******************************************
C                    * ROUTINE NAME - LINE WEIGHT COMPUTATION   *
C                    * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                    * PROGRAMMER   - R, BOLLINGER 1943 102 26933 *
C                    * DATE CODED   - 3/17/70                   *
C                    * REVISED      - 12-17-71    (RFH)
C                    ******************************************
C
      SUBROUTINE LWEGHT(IDX,LDV)
C
C        ***** EXPLANATION OF THE CALLING SEQUENCE
C              *
C        ***** IDX - INDEX OF THE CONFIGURATION TABLE.
C
      LOGICAL DIAG
C
C
      INCLUDE CCNFIG
      INCLUDE CMATRL
      INCLUDE CONST
      INCLUDE TABLOK
C
      IF(CMTYPE.LE.5) GO TO 5
      IF(CMTYPE.EQ.6) GO TO 50
      IF(CMTYPE.EQ.7) GO TO 60
C
    5 CONTINUE
C
C        ***** LOOK UP FTU (SI) FOR LINE
C
      IF(DIAG(0,6HLWEGHT)) WRITE (6,6000) RHOL(CMTYPE),RHOI(CMTYPE)
C
      CALL FINTAB (NTBID(18)+CMTYPE)
      XTAB(1)=TEMP(IDX)
      SI = MIPE(1,XTAB)
C
C        ***** COMPUTE THE THICKNESS FOR FTU
C
      THKL = PRES(IDX)*DIAM(IDX)*2.5/(2.0*SI)
C
      JKM = 0
      IF (PRES(IDX) .GE. 1000.) JKM = 5
      IF (PRES(IDX) .GE. 3000.) JKM = 10
      IF (THKL .LE. MINTHK(CMTYPE+JKM)) THKL = MINTHK(CMTYPE+JKM)
C
C        ***** COMPUTE THE WGT/FT FOR THE TUBING OR PIPE MATERIAL
C
      WGTFT = PI * DIAM(IDX) * THKL * RHOL(CMTYPE)/144.0
C
      GO TO (71,72,73,74,75),LDV
C
C        ***** COMPUTE EQUIVALENT LENGTH OF FITTING
C
C           ** FOR A 4-WAY TEE
C
   72 FLOD = 4.75 * DIAM(IDX)
      GO TO 80
```

```
*********     LWEGHT    *********

      C
      C          ** FOR A 3-WAY TEE
      C
         73 FLOD = 3.75 * DIAM(IDX)
            GO TO 80
      C
      C          ** FOR A 90 DEG.ELBOW
      C
         74 FLOD = 2.75 * DIAM(IDX)
            GO TO 80
      C
      C          ** FOR A 45 DEG.ELBOW
      C
         75 FLOD = 1.75 * DIAM(IDX)
            GO TO 80
      C
      C        ***** COMPUTE THE WEIGHT OF A FITTING
      C
         80 WEIGHT(IDX) = WGTFT * (FLOD/12.) * 1.25
      C
            RETURN
      C
      C        ***** COMPUTE THE WEIGHT OF A LINE
      C
         71 WEIGHT(IDX) = WGTFT * LOD(IDX)/12.
      C
      C        ***** COMPUTE THE WEIGHT OF THE INSULATION
      C
            WI(IDX) = PI * ITHICK(IDX) * LOD(IDX) * RHOI(CITYPE) * (DIAM(IDX)
           !          + ITHICK(IDX)/2.0)/1728.0
      C
            IF(DIAG(1,6HLWEGHT)) WRITE(6,6000) THKL,SI,WGTFT
            RETURN
      C
         50 CONTINUE
      C
      C        ***** COMPUTE WEIGHT OF VACUUM JACKETED CRES LINE  (321/347)
      C
            AYE1 = 0.217684
            BEE1 = -6.69016E-03
            WGTFT = 1.0/((AYE1)+(BEE1)*DIAM(IDX))
            WEIGHT(IDX) = WGTFT * LOD(IDX)/12.
      C
            IF(DIAG(1,6HLWEGHT))WRITE (6,6000) AYE1,BEE1,DIAM(IDX),WGTFT,
           !                          LOD(IDX)
            RETURN
      C
         60 CONTINUE
      C
      C        ***** COMPUTE WEIGHT OF VACUUM JACKETED ALUMINUM LINE (2219)
      C
            AYE2 = 0.559277
            BEE2 = -2.00888E-02
            WGTFT = 1.0/((AYE2)+(BEE2)*DIAM(IDX))
            WEIGHT(IDX) = WGTFT * LOD(IDX)/12.
      C
            IF(DIAG(1,6HLWEGHT)) WRITE(6,6000) AYE2,BEE2,DIAM(IDX),WGTFT,
           !                         LOD(IDX)
```

******** LWEGHT ********

```
C
      RETURN
C
 6000 FORMAT('+',14X,5E15.8)
      END
```

******** SUBRØUTINE MATHAX

```
      SUBROUTINE PUMPEF (KUP)
C
      INCLUDE CIOUNT
      INCLUDE SPUMP,LIST
      INCLUDE TABLOK
C
      DIMENSION PCRIT(2),CRTMP(2)
      DIMENSION CT1(2),CT2(2),CT3(2),CT4(2),CT5(2),
     1          CH1(2),CH2(2),CH3(2),CH4(2),CH5(2)
C
      DATA PCRIT /731.379,187.506/ CRTMP / 278.237,59.956/
      DATA CT1 /.1,.3/ CT2 /.8,.82/ CT3 /.48,.521/ CT4 / .372,.414/
     1 CT5 /.2185,.2685/ CH1 /.31,.212/ CH2 /.81,.73/ CH3 /.19,.498/
     2 CH4 /.0016,.000123/ CH5 /0.,.00513/
C
 6001 FORMAT ('0'20X'*** TCR(MIN) > TCR(MAX) CHANGE DESIGN POINT ***')
C
C
      DO 100 K=1,KUP
      NSG = NSG + DNS/XNS
C
C                    CALC. SPECIFIC SPEED (NSS)
C
      NSS = NSG*XNS
C
C                    FIND HEAD COEF.  PSI = F(NSS)
C
      CALL FINTAB (NTBID(19))
      PSI = MIPE (1,NSS)
C
C                   FIND ADIABATIC EFF. NUZ = F(NSS)
C
      CALL FINTAB (NTBID(20))
      NUZ = MIPE (1,NSS)
C
C                   CALC. IMPELLER TIP SPEED (U)
C
      U = SQRT (32.2*H/PSI)
C
C                   CALC. IMPELLER DIAMETER
C
      DI = 229*U /NSG
C
C                   FIND EFF. QUOTIENT
C
      CALL FINTAB (NTBID(21))
      EFFQ = MIPE (1,DI)
C
C                   CALC. PUMP HYDRAULIC EFF
C
      NU(K) = EFFQ*NUZ
  100 CONTINUE
C
      RETURN
C
C                   USED BY THE HEATEX ROUTINE
C
```

********   MATHAX   ********

```
      ENTRY TCRCAL (THA,THB,TCA,TCB,TCRMN,TCRMX,IERR)
C                    CALCULATE TCR MIN
      TCRA = (THA-1800.)/(1800.-TCA)
      TCRB = (THB-1800.)/(1800.-TCB)
      IF (TCRA .LT. 0.) TCRA = 0.
      IF (TCRB .LT. 0.) TCRB = 0.
      TCRMN = TCRA
      IF (TCRA .LT. TCRB) TCRMN = TCRB
C                    CALCULATE TCR MAX.
      TCRA = (THA-550.)/(550.-TCA)
      TCRB = (THB-550.)/(550.-TCB)
      IF (TCRA .LT. 0.) TCRA = 5000.
      IF (TCRB .LT. 0.) TCRB = 5000.
      TCRMX = TCRA
      IF (TCRA .GT. TCRB) TCRMX = TCRB
      IF (TCRMX .EQ. 5000.) IERR = 1
      IF (TCRMN .LT. TCRMX) GO TO 200
      WRITE (IOT,6001)
      IERR = 1
  200 RETURN
C
C                    USED BY THE HEATEX ROUTINE
C
      ENTRY ENTHOH (PE,TE,KG,ENTH)
      IF (KG .GT. 1) GO TO 300
      ENTH = OXENTH (PE,TE)
      RETURN
  300 ENTH = HYENTH (PE,TE)
      RETURN
C
C                    FIND DENSITY OF O2 OR H2  (LIQ. OR GAS)
C
      ENTRY FDNSTY (IG,TEMP,PRES,RHO)
C
      IF (PRES .GT. PCRIT(IG) .AND. TEMP .GT. CRTMP(IG)) GO TO 400
C                    LIQUID STATE
      CALL RHOLIQ (TEMP,IG,RHO)
      RETURN
C
      ENTRY GSDNST (IG,TEMP,PRES,RHO)
C                    DENSITY AND Z RETURNED
      ENTRY GSZDNS (IG,TEMP,PRES,RHO,Z)
C                    GASEOUS STATE
  400 CALL ZFIND (TEMP,PRES,IG,Z)
      RHO = 144.*PRES / (Z*FINDR(IG)*TEMP)
      RETURN
C
C                    USED BY THE HEATEX ROUTINE
C
      ENTRY TCRCLC  (IGAS,FDPC,FPCIN,DPH,PHIN,TCR)
C
      FDPOPC = FDPC / FPCIN
      DPOPH = DPH / PHIN
C                    CALC. TCR FOR O2 OR H2
  490 TCR = CT1(IGAS)*(FPCIN**CT2(IGAS))/(PHIN**CT3(IGAS))
     1         *(FDPOPC**CT4(IGAS))/(DPOPH**CT5(IGAS))
      GO TO 500
C
```

```
********    MATHAX    ********

            ENTRY WOUACL (WOUA)
      C
      C                    CALC.  (W/UA)
            ALPHW = CW4(IGAS)*FPCIN + CW5(IGAS)*PHIN
      C
            WOUA = CW1(IGAS)/((FPCIN**CW2(IGAS))*(PHIN**CW3(IGAS)))*EXP(ALPHW)
      C                    IS IT HYDROGEN
            IF (IGAS .EQ. 1) GO TO 500
      C                    FOR H2 ONLY
            WOUA = WOUA / ((FDPOPC**0.248)*(DPOPH**0.1656))
        500 CONTINUE
            RETURN
      C
            ENTRY TCRLOW (FPCOUT,IGO)
      C                    LOWER TCR VALUE
            FDPOPC = FDPOPC / 2.
            IF (FDPOPC .GT. 0.001) GO TO 510
            IGO = 2
            FDPOPC = 0.001
            GO TO 520
      C
            ENTRY TCRRAZ (FPCOUT,IGO)
      C                    RAISE TCR VALUE
            FDPOPC = 2.0 * FDPOPC
            IF (FDPOPC .LT. 0.20) GO TO 510
            IGO = 2
            FDPOPC = 0.20
            GO TO 520
        510 IGO = 1
        520 FPCIN = FPCOUT / (1. - FDPOPC)
            FDPC = FPCIN - FPCOUT
            GO TO 490
      C
            END
```

```
C                   * * * * * * * * * * * * * * * * * * * * * * * * *
C                   * ROUTINE NAME - MULTI-TABLE INTERPOLATION      *
C                   *                AND POLYNOMIAL EVALUATION       *
C                   * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                   * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                   * DATE CODED   - 3/3/70                      *
C                   * REVISED      - SEPTEMBER 3,1971            *
C                   * PROGRAMMER   - J. MCKAY  19-43  201 X45178 *
C                   * * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION MIPE(NL,XVAL)
C
C         ***** EXPLANATION OF THE CALLING SEQUENCE
C
C                   NL   - THE NUMBER OF LEVELS OF INTERPOLATION OR ONE
C                          LESS THAN THE NUMBER OF DIMENSIONS OR THE NUMBER
C                          OF VALUES IN THE ARRAY XVAR.
C
C                   XVAR - AN ARRAY OF INDEPENDENT VARIABLES ON WHICH
C                          TO INTERPOLATE OR EVALUATE POLYNOMIALS.
C
      LOGICAL DIAG
C
      REAL MIPE
C
      INCLUDE CIOUNT
      INCLUDE CKEYS
      INCLUDE CTAB
C
      DIMENSION  XVAL(6),YVAL(32),KNT(5),L(5),NTD(6),TAB1(6,5),
     1  ITAB1(6,5)
C
      EQUIVALENCE (TAB1,ITAB1,TAB,ITAB)
C
C         ***** LOCATE THE POSITION IN EACH OF THE FIRST NL-1 TABLES OF
C         ***** THE INDEPENDENT VARIABLE AND STORE THE INDEXES IN KNT.
C
      MPX = 1
    1 CONTINUE
      IF (DIAG(0,6HMIPE  )) WRITE (IOT,6000) NL,(TAB(1,I),I=1,5),
     1                                     (XVAL(I),I=1,NL)
      IF(NL.GT.1) GO TO 5
      CALL LOCAT (MPX)
      NLTBL = 1
      CALL TEL(XVAL(1),YVAL(1))
      GO TO 220
    5 NLM1 = NL - 1
      DO 70 I1 = 1,NLM1
      NTAB1 = ITAB1(1,I1)
      IF(XVAL(I1) - TAB1(2,I1)) 50,50,10
   10 IF(XVAL(I1) - TAB1(NTAB1+1,I1)) 20,60,60
C
   20 DO 30 I2 = 1,NTAB1
      I3 = I2
      IF(XVAL(I1) - TAB1(I2+1,I1)) 40,40,30
   30 CONTINUE
C
```

```
   40 KNT(I1) = I3 - 1
      GO TO 70
C
   50 KNT(I1) = 1
      GO TO 70
C
   60 KNT(I1) = NTAB1 - 1
   70 CONTINUE
C
C        ***** COMPUTE THE TABLE NUMBER OF THE XY TABLES FOR
C        ***** INTERPOLATION.
C
      NT = 2**NLM1
      NLTBL = NT
      DO 110 I1=1,NL
  110 NTD(I1) = NT / 2**(I1-1)
C                                 CALC. INTERPOLATION LEVEL COEF.
      L(NLM1) = 1
      IF (NLM1 .EQ. 0) GO TO 125
      DO 120 I1=NLM1,2,-1
  120 L(I1-1) = L(I1) * ITAB1(1,I1)
C                                 COMPUTE THE ID NO. OF THE X-Y SUB-TABLES
C                                 USED FOR INTERPOLATION
  125 DO 140 I1=1,NT
      KTB = 1
      DO 130 I2=1,NLM1
  130 KTB = KTB + L(I2)* (KNT(I2)-1+MOD(I1-1,NTD(I2))/NTD(I2+1))
C                                 LOOK UP SUBTABLE LOCATION
      CALL LOCAT (MPX*KTB)
C                                 COMPUTE THE YVAL FOR XY-CURVES OR
C                                 POLYNOMIALS
      CALL TEL (XVAL(NL),YVAL(I1))
      IF (KEY2 .EQ. KEY1) RETURN
  140 CONTINUE
      IF (DIAG(2,6HMIPE  )) WRITE (IOT,6000) KNT,NT ,(YVAL(I),I=1,NT)
C
C        ***** COMPUTE THE FINAL VALUE OF YVAL
C
      DO 210 I1 = NLM1,1,-1
      KYVAL = 1
      IDX1 = KNT(I1) + 1
      FAC = (XVAL(I1) - TAB1(IDX1  ,I1))/(TAB1(IDX1+1,I1)-TAB1(IDX1,I1))
C
      DO 200 I2 = 1,NT,2
      YVAL(KYVAL) = (YVAL(I2+1) - YVAL(I2))*FAC + YVAL(I2  )
      KYVAL = KYVAL + 1
  200 CONTINUE
C
      NT = 2**(I1-1)
  210 CONTINUE
C
C        ***** SET THE ANSWER AND RETURN TO THE CALLING PROGRAM
C
  220 MIPE = YVAL(1)
      IF (DIAG(1,6HMIPE  )) WRITE (IOT,6000) NTD,YVAL(1)
      RETURN
C
C                                 RVMIPE CAUSES INDEP AND DEP VAR TO SWITCH
```

```
*********   MIPE   *********

      ENTRY RVMIPE (NL,XVAL)
C
      MPX = -1
      GO TO 1
C
 6000 FORMAT ('+'14X,I7,5I15/(30X5E15.6))
      END
```

********* FUNCTIØN NIENTH

```
      FUNCTION NIENTH(PRES,TEMP)
C
C     ROUTINE CALLS STEWARTS O2-N2 PROPERTIES PROGRAM
C
      REAL NIENTH
C
      CALL ONPROP(TEMP,PRES,DX,ENTH18,SX,UX,ZX,18)
      NIENTH = ENTH18
      RETURN
      END
```

******** SUBRØUTINE ØNPRØP

```
      SUBROUTINE ONPROP(TEMP,PRES,DENS,ENTH,ENTR,ENERG,ZEE,NGAS)
C
      COMMON /METH/ M
C
C     INPUT TO THIS SUBROUTINE MUST BE IN BRITISH UNITS
C
C
      TB = TEMP
      PB = PRES
      DB = DENS
      M = 1
C
      IF(NGAS.EQ.1) KF = 1
      IF(NGAS.EQ.18) KF = 2
C
C     KF = 1  CALL IN OXYGEN PARAMETERS
C     KF = 2  CALL IN NITROGEN PARAMETERS
C
      IF(KF.EQ.1) CALL DATAO2
      IF(KF.EQ.2) CALL DATAN2
C
      IF((TB.GT.0.000).AND.(PB.GT.0.000)) GO TO 5
      IF((TB.GT.0.000).AND.(DB.GT.0.000)) GO TO 10
C
    5 CALL PROPB(TB,PB,DB,1,HB,SB,UB,ZB)
      DENS = DB
      ENTH = HB
      ENTR = SB
      ENERG = UB
      ZEE = ZB
C
      RETURN
C
   10 CALL PROPB(TB,PB,DB,2,HB,SB,UB,ZB)
      PRES = PB
      ENTH = HB
      ENTR = SB
      ENERG = UB
      ZEE = ZB
C
      RETURN
C
      END
```

********* SUBRØUTINE ØTRTNS

```
C
C                     GENERAL OUTPUT ROUTINES
C
      SUBROUTINE OTPHEX
C
      LOGICAL  JP,PAGE
C
      INCLUDE CACCUM
      INCLUDE CAPU
      INCLUDE CCNFIG
      INCLUDE CECLSS
      INCLUDE CENG
      INCLUDE CFLRAT
      INCLUDE CFUEL
      INCLUDE CHEX
      INCLUDE CHSORC
      INCLUDE CIOUNT
      INCLUDE CNAMES
      INCLUDE CPUMP
      INCLUDE CTANK
      INCLUDE CTURBN
C
      DATA  IDMHX/ MHX/
C
 6010 FORMAT(1H0,T39,'*** SUMMARY OF COMPUTED HEAT EXCHANGER CHARACTERIS
     1TICS ***'//T59,'FOR UNITS',T73,A6,T93,A6)
 6020 FORMAT(1H0,T29,'*** SUMMARY OF COMPUTED HEAT EXCHANGER-GAS GENERAT
     1OR CHARACTERISTICS ***')
 6025 FORMAT(1H0,T36,'*** SUMMARY OF COMPUTED WASTE HOT GAS UTILIZATION
     1PARAMETERS ***')
 6026 FORMAT(1H0,T36,'*** SUMMARY OF COMPUTED FREON COOLANT UTILIZATION
     1PARAMETERS ***')
 6030 FORMAT(1H0,T35,'*** SUMMARY OF COMPUTED HEAT EXCHANGER CHARACTERIS
     1TICS -- CONTINUED ***')
 6040 FORMAT(1H0,T40,'*** SUMMARY OF COMPUTED PUMP CHARACTERISTICS FOR T
     1HE SYSTEM ***')
 6050 FORMAT(1H0,T36,'*** SUMMARY OF COMPUTED TURBINE CHARACTERISTICS FO
     1R THE SYSTEM ***')
 6060 FORMAT(1H0,T36,'*** SUMMARY OF COMPUTED TURBINE GAS GENERATOR CHAR
     1ACTERISTICS ***')
 6065 FORMAT(///T46,'*** ECLSS ENERGY REQUIREMENTS SUMMARY *** ')
 6070 FORMAT(1H0,T46,'*** INITIAL TANK SIZING CALCULATIONS ***')
 6080 FORMAT(1H0,T47,'*** FINAL TANK SIZING CALCULATIONS ***')
 6090 FORMAT(1H0,T47,'*** ACCUMULATOR SIZING CALCULATIONS ***')
 6100 FORMAT(1H0,T39,'*** TANK PROPELLANT ACQUISITION DEVICE COMPUTATION
     1***')
 6110 FORMAT(1H0,T51,'*** COMPONENT WEIGHT SUMMARY ***'// T27,'... OXIDY
     1ZER ...'T87,'... FUEL ...' / T31,'---------'T91,'----'/ T38,'COMPON
     2ENT'T52,'INSULATION'T98,'COMPONENT'T112,'INSULATION' / T11,'COMPON
     3ENT'T27,'CODE'T38,'WT. (LBS)'T52,'WT. (LBS)'T71,'COMPONENT'T87,
     4'CODE'T98,'WT. (LBS)'T112,'WT. (LBS)'/)
 6112 FORMAT(1H0,T47,'*** COMPONENT WEIGHT SUMMARY TOTALS ***')
 6120 FORMAT(1H0,T52,'*** COMPUTED FLOWRATE DATA ***')
 6130 FORMAT(1H0,T47,'*** INITIAL APU PROGRAM CALCULATIONS ***'//
     1 T40,'PERCENT'T61,'TOT. FLOWRATE'T85,'EXHAUST TEMP.' /
     2 T25,'CYCLE'T42,'H.P.'T63,'(LB./MIN.)'T87,'(DEG. - R)'/)
 6132 FORMAT(T46,'*** APU MIXTURE RATIO IS NOW SET AT',F5.2,' ***'/)
```

******** OTRTNS ********

```
 6140 FORMAT(24XI4,E20.6,2E24.6)
 6150 FORMAT(1H0,T51,'*** APU CALCULATIONS - CONT. ***')
 6160 FORMAT(T44,'*** APU MIXTURE RATIO IS NOW SET AT'F5.2,' ****'///
     1           T47,'*** APU SUPERCRITICAL CALCULATIONS ***')
 6161 FORMAT(4XI2,8E15.6)
 6162 FORMAT(1H0  T14,'CONDITIONING GAS REQD'T44, 'WGT. CONDITIONING GAS
     1'              T74,'CONDITIONING GAS REQD'T104,'WGT. CONDITIONING GAS
     2'/T2,'CYCLE'T15,'ACCUMULATOR TO APU'   T45, 'ACCUMULATOR TO APU'
     3               T75,'TANK TO ACCUMULATOR'  T105,'TANK TO ACCUMULATOR'
     4 /             T6,4(8X'OXYGEN'8X'HYDROGEN')/)
 6164 FORMAT(1H0  T14,'CONDITIONING GAS REQD'T44, 'WGT. CONDITIONING GAS
     1'              T74,'FLOW TO ACCUMULATOR'  T101,'EXHAUST PROD.'
     2 /T2,'CYCLE'T21,'FOR TANK' T51,'FOR TANK'T78,'(EACH CYCLE)'
     3               T103,'SURPLUS'/T6,3(8X'OXYGEN'8X'HYDROGEN')/)
 6166 FORMAT(1H0  T16,'REFERENCE FLOW TO'    T44, 'TEMPERATURE OF FLUID'
     1               T75,'PERCENT OF USABLE'    T104,'HEAT XFR. INTO TANK'
     2 /T2,'CYCLE'T15,'SUPPLEMENTAL G. G.'   T46, 'IN STORAGE TANK'
     3               T76,'FLUID WITHDRAWN'      T108,'(EACH CYCLE)'
     4 /             T6,4(8X'OXYGEN'8X'HYDROGEN')/)
 6168 FORMAT(1H0  T16,'ENERGY DERIVATIVE'    T44, 'COR. COND. GAS REQD.'
     1               T74,'COR. WGT. COND. GAS'
     2 /T2,'CYCLE'T45,'TANK TO ACCUMULATOR'  T74, 'TANK TO ACCUMULATOR'
     3 /             T6,3(8X'OXYGEN'8X'HYDROGEN')/)
 6170 FORMAT(1H0  T44,'COR. COND. GAS REQD.' T74, 'COR. WGT. COND. GAS'
     1/T32,'CYCLE'T50,'FOR TANK' T80,'FOR TANK'
     2 /             T36,2(8X'OXYGEN'8X'HYDROGEN')/)
 6171 FORMAT(34XI2,6E15.6)
 6172 FORMAT(T44,'*** APU MIXTURE RATIO IS NOW SET AT'F5.2,' ****'///
     1           T48,'*** APU SUBCRITICAL CALCULATIONS ***')
 6174 FORMAT(1H0  T14,'CONDITIONING GAS REQD'T44, 'WTG. CONDITIONING GAS
     1'              T74,'CONDITIONING GAS REQD'T104,'WTG. CONDITIONING GAS
     2'/T2,'CYCLE'T15,'ACCUMULATOR TO APU'   T45, 'ACCUMULATOR TO APU'
     3               T75,'PUMP TO ACCUMULATOR'  T105,'PUMP TO ACCUMULATOR'
     4 /             T6,4(8X'OXYGEN'8X'HYDROGEN')/)
C
      NIENTH = 0.0
C                   DO OUTPUT FOR ALL HEAT EXCHANGERS
C
      IDM2 = 2*IDMHX
      IDM3 = 3*IDMHX
      IDM6 = 6*IDMHX
      DO 140 JX=1,NUMHEX
      KVL = 0
      IF (NSUK(JX,1).EQ.0) KVL = KVL + 1
      IF ( NSUK(JX,2).EQ.0) KVL = KVL + 2
      MVL = 3 - KVL
      IF (MVL.EQ.0) GO TO 140
      IF (PAGE( 0)) WRITE (IOT,6010) UCODE(JX,1),UCODE(JX,2)
      JP = PAGE (17)
      CALL SPACE
      CALL OUTPA (5,LHX3,MVL,JFLUID,JFLUID(1,2))
      CALL SPACE
C
      DO 100 I2=1,11
      JSM = (I2-1)*IDM2 + JX
      JLM = JSM + IDMHX
C
      CALL OUTPF (4,LHX2(1,I2),MVL,UOTHX(JSM),UOTHX(JLM))
```

```
  100 CONTINUE
      J1 = NSSK(JX,1) - 1
      J2 = NSSK(JX,2) - 1
      MXNSU  = NSUK(JX,1)
      IF (MXNSU .LT. NSUK(JX,2)) MXNSU = NSUK(JX,2)
C
      DO 120 I2=1,MXNSU
      IF (PAGE(11)) WRITE (IOT,6030)
      J1 = J1 + 1
      K1 = 1
      IF (J1 .EQ. 1 .AND. IISU(JX,1) .GT. 7) K1 = 0
      J2 = J2 +1
      K2 =1
      IF (J2 .EQ. 1 .AND. IISU(JX,2) .GT. 7) K2 = 0
      NVL = 0
      IF (J1 .LE. NSSK(JX,1)+NSUK(JX,1)-1) NVL = NVL+1
      IF (J2 .LE. NSSK(JX,2)+NSUK(JX,2)-1) NVL = NVL+2
      CALL SPACE
      CALL OUTPA (4,LHX1(1,10),NVL,UNAM(1,J1+K1),UNAM(1,J2+K2))
      CALL SPACE
      DO 110 I3=1,8
      JSM = (I3-1)*IDM6 + (J1-1)*IDMHX + JX
      JLM = (I3-1)*IDM6 + (J2-1)*IDMHX + IDM3 + JX
C
      CALL OUTPF (4,LHX1(1,I3),NVL,UOTHY(JSM),UOTHY(JLM))
  110 CONTINUE
  120 CONTINUE
      IF (PAGE( 2)) WRITE (IOT,6030)
      CALL SPACE
C
      CALL OUTPF (4,LHX1(1,9),MVL,WHXTOT(JX,1),WHXTOT(JX,2))
      JP = PAGE (10)
C
      JSTYP = HSTYPE(JX,2)
      IF(JSTYP.EQ.2) GO TO 125
C
      WRITE (IOT,6020)
      CALL SPACE
C
      CALL OUTPA (5,LHS1,MVL,JFLUID,JFLUID(1,2))
      CALL SPACE
      CALL OUTPF (5,LHS1(1,2),MVL,WDOTH(JX,1),WDOTH(JX,2))
      CALL OUTPF (5,LHS1(1,3),MVL,WGGFX(JX,1),WGGFX(JX,2))
      CALL OUTPF (5,LHS1(1,4),MVL,HSWGHT(JX,1),HSWGHT(JX,2))
      CALL SPACE
      CALL OUTPF (5,LHS1(1,6),MVL,HXASSY(JX,1),HXASSY(JX,2))
C
      IF (JSTYP .EQ. 1) GO TO 140
C
  125 CONTINUE
      WRITE (IOT,6025)
      CALL SPACE
C
      CALL OUTPA (5,LHS2(1,6),MVL,JFLUID,JFLUID(1,2))
      CALL SPACE
      CALL OUTPF (5,LHS2(1,1),MVL,HSGCPE(JX,1),HSGCPE(JX,2))
      CALL OUTPF (5,LHS2(1,2),MVL,HSQREQ(JX,1),HSQREQ(JX,2))
      CALL OUTPF (5,LHS2(1,3),MVL,HSGTOT(JX,1),HSGTOT(JX,2))
```

```
C
  140 CONTINUE
      IF (PAGE ( 5)) WRITE (IOT,6030)
      CALL SPACE
      CALL OUTPF (5,LHS1(1,5),3,WGGTOT,WGGTOT(2))
C
      CALL SPACE
      CALL OUTPF (5,LHS2(1,4),3,QTREQ(1),QTREQ(2))
      CALL OUTPF (5,LHS2(1,5),3,HFTOT(1),HFTOT(2))
      RETURN
C
      ENTRY OTPHXF
C
C               *** OUTPUT FUEL CELL HEAT EXCHANGER PARAMETERS
C
      IDM2 = 2*IDMHX
      DO 141 JX = 1,NUMHEX
      IF(PAGE(0)) WRITE(IOT,6010) UCODE(JX,1),UCODE(JX,2)
      JP =  PAGE (17)
      MVL = 3
      CALL SPACE
      CALL OUTPA (5,LHX3,MVL,JFLUID,JFLUID(1,2))
      CALL SPACE
C
      DO 142 I2 = 1,8
      JSM = (I2-1) * IDM2 + JX
      JLM = JSM + IDMHX
      CALL OUTPF (4,LHX2(1,I2),MVL,UOTHX(JSM),UOTHX(JLM))
  142 CONTINUE
      CALL SPACE
      CALL OUTPF (4,LHX1(1,9),MVL,WHXTOT(JX,1),WHXTOT(JX,2))
C
      WRITE (IOT,6026)
      CALL SPACE
      CALL OUTPA (5,LHS2(1,6),MVL,JFLUID,JFLUID(1,2))
      CALL SPACE
      CALL OUTPF (5,LHS2(1,1),MVL,HSGCPE(JX,1),HSGCPE(JX,2))
      CALL OUTPF (5,LHS2(1,2),MVL,HSQREQ(JX,1),HSQREQ(JX,2))
      CALL OUTPF (5,LHS2(1,3),MVL,HSGTOT(JX,1),HSGTOT(JX,2))
  141 CONTINUE
C
      CALL SPACE
      CALL OUTPF (5,LHS2(1,4),3,QTREQ(1),QTREQ(2))
      CALL OUTPF (5,LHS2(1,7),3,WDTFMX   ,WDTFMX   )
C
      RETURN
C
C
      ENTRY OTPHXE
C
C               *** OUTPUT THE ECLSS HEAT EXCHANGER PARAMETERS
C
      IDM2 = 2*IDMHX
      DO 143 JX = 1,NUMHEX
      IF(PAGE(0)) WRITE(IOT,6010) UCODE(JX,1),UCODE(JX,2)
      JP =  PAGE (17)
      MVL = 3
      CALL SPACE
```

```
      CALL OUTPA(5,LHX3,MVL,JFLUID,JFLUID(1,3))
      CALL SPACE
      DO 144 I2 = 1,2
      JSM = (I2-1) * IDM2 + JX
      JLM = JSM + IDMHX
      CALL OUTPF (4,LHX2(1,I2),MVL,UOTHX(JSM),UOTHX(JLM))
  144 CONTINUE
C
      CALL OUTPF(4,LHX2(1,4),MVL,WDOTCF(JX,1),WDOTCF(JX,2))
      CALL OUTPF (4,LHX2(1,12),MVL,UOA(JX,1),UOA(JX,2))
      CALL OUTPF (4,LHX2(1,13),MVL,DH(JX,1),DH(JX,2))
      CALL OUTPF (4,LHX2(1,14),MVL,HLNGTH(JX,1),HLNGTH(JX,2))
      CALL SPACE
      CALL OUTPF(4,LHX1(1,3),MVL,HXCDLP(JX,1),HXCDLP(JX,2))
      CALL SPACE
      CALL OUTPF (4,LHX1(1,9),MVL,WHXTOT(JX,1),WHXTOT(JX,2))
      CALL SPACE
      CALL OUTPF (5,LHS2(1,2),MVL,HSQREQ(JX,1),HSQREQ(JX,2))
  143 CONTINUE
C
      RETURN
C
      ENTRY OPTPOW
C
C             *** OUTPUT THE POWER REQUIREMENTS SUMMARY
C
      WRITE (IOT,6065)
      CALL SPACE
      CALL OUTPF1 (5,LHS2(1,8),HWTOMX)
      CALL OUTPF1 (5,LHS2(1,9),HWTNMX)
      CALL OUTPF1 (5,LHS2(1,10),TWTOMX)
      CALL OUTPF1 (5,LHS2(1,11),TWTNMX)
      CALL SPACE
      CALL OUTPF1 (5,LHS2(1,12),TOTWMX)
      CALL SPACE
      CALL OUTPF1 (5,LHS2(1,13),TOTWAT)
      CALL SPACE
      CALL OUTPF1 (5,LHS2(1,14),TOTPOW)
C
      RETURN
C
C
      ENTRY OTPPMP
C                     OUTPUT PUMP PARAMETERS
      IF (PAGE( 0)) WRITE (IOT,6040)
      JP = PAGE (21)
      CALL SPACE
      CALL OUTPA (4,LPP5,3,JFLUID,JFLUID(1,2))
      CALL SPACE
      DO 200 I=1,6
      I2 = 2*(I-1)+1
      CALL OUTPF (3,LPP1(1,I),3,UOTP1(I2),UOTP1(I2+1))
  200 CONTINUE
      CALL SPACE
      CALL OUTPI (4,LPP2,3,PSTAGE,PSTAGE(2))
      DO 210 I=1,6
      I2 = 2*(I-1)+1
      CALL OUTPF (3,LPP3(1,I),3,UOTP2(I2),UOTP2(I2+1))
```

```
  210 CONTINUE
      CALL SPACE
      CALL OUTPI (4,LPP4,3,JJOPT,JJOPT(2))
      RETURN
C
      ENTRY OTPTRB
C                     DO OUTPUT FOR TURBINES
      JP = PAGE (19)
      WRITE (IOT,6050)
      CALL SPACE
      CALL OUTPA (4,LTBN1,3,JFLUID,JFLUID(1,2))
      CALL SPACE
      DO 300 I=1,6
      I2 = 2*(I-1)+1
      CALL OUTPF (5,LTBN2(1,I),3,UOTB1(I2),UOTB1(I2+1))
  300 CONTINUE
      WRITE (IOT,6060)
      CALL SPACE
      CALL OUTPA (5,LHS1,3,JFLUID,JFLUID(1,2))
      CALL SPACE
      DO 310 I=1,3
      I2 = 2*(I-1)+1
      CALL OUTPF (5,LHS1(1,I+1),3,UOTB2(I2),UOTB2(I2+1))
  310 CONTINUE
      RETURN
C
      ENTRY OTPTSZ (IFLG)
C                     OUTPUT TANK SIZING PARAMETERS
      JP = PAGE (0)
      JP = PAGE (17)
      IF (IFLG .EQ. 2) GO TO 400
      WRITE (IOT,6070)
      GO TO 410
  400 WRITE (IOT,6080)
  410 CALL SPACE
      CALL OUTPA (1,6H      ,3,JFLUID,JFLUID(1,2))
      CALL SPACE
      CALL OUTPI (3,LTZ1(1,1),3,NOP   ,NOP   (2,1))
      CALL OUTPI (3,LTZ1(1,2),3,SMTYPE,SMTYPE(2,1))
      IF (IFLG .EQ. 2)
     1CALL OUTPI (3,LTZ1(1,3),3,SITYPE,SITYPE(2,1))
      CALL OUTPF (3,LTZ1(1,4),3,WPTOT ,WPTOT (2)  )
      CALL OUTPF (3,LTZ1(1,5),3,SVLFLD,SVLFLD(2)  )
      CALL OUTPF (3,LTZ1(1,14),3,TCYHT,TCYHT (2)  )
      CALL OUTPF (3,LTZ1(1,6),3,SMDIAM,SMDIAM(2,1))
      CALL OUTPF (3,LTZ1(1,7),3,TSA   ,TSA   (2,1))
      CALL OUTPF (3,LTZ1(1,8),3,SVOL  ,SVOL  (2,1))
      CALL OUTPF (3,LTZ1(1,9),3,TWT   ,TWT   (2,1))
      IF (IFLG .EQ. 2) GO TO 420
      CALL OUTPF (3,LTZ1(1,12),3,SHRATE,SHRATE(2,1))
      GO TO 430
  420 CALL OUTPF (3,LTZ1(1,10),3,SITHIK,SITHIK(2,1))
      CALL OUTPF (3,LTZ1(1,11),3,TIWT  ,TIWT  (2,1))
C
  430 RETURN
C
      ENTRY OTPACC
C                     OUTPUT ACCUMULATOR DATA
```

```
      IF (PAGE( 0)) WRITE (IOT,6090)
      JP = PAGE (16)
      CALL SPACE
      CALL OUTPA (1,6H      ,3,JFLUID,JFLUID(1,2))
      CALL SPACE
      DO 500 I=1,3
      I2 = 2*(I-1)+1
  500 CALL OUTPI (3,LTZ1(1,I),3,EQAC(I2),EQAC(I2+1))
      CALL OUTPF (3,LTZ1(1,14),3,ACYHT,ACYHT(2))
      DO 510 I=6,11
      I2 = 2*(I-3)+1
  510 CALL OUTPF (3,LTZ1(1,I),3,EQAC(I2),EQAC(I2+1))
      CALL OUTPF (3,LTZ1(1,13),3,WGRACC,WGRACC(2))
      RETURN
C
      ENTRY OTPACQ
C                         OUTPUT ACQUISITION DEVICE DATA
      IF ( PAGE( 0)) WRITE (IOT,6100)
      JP = PAGE ( 9)
      CALL SPACE
      CALL OUTPA (1,6H      ,3,JFLUID,JFLUID(1,2))
      CALL SPACE
      I1 = SATYPE
      I2 = SATYPE(2)
      CALL OUTPA (3,LTZ3(1,1),3,LTZ2(1,I1),LTZ2(1,I2))
      CALL OUTPF (3,LTZ3(1,2),3,WTACQ,WTACQ(2,1))
      CALL OUTPF (3,LTZ3(1,3),3,WLR,WLR(2))
      CALL OUTPF (3,LTZ3(1,4),3,WLRT,WLRT(2,1))
      RETURN
C
      ENTRY OTPWSM
C                         OUTPUT WEIGHT SUMMARY DATA
      IF (PAGE( 0)) WRITE (IOT,6110)
      JP = PAGE ( 8)
      I1 = IOSTT
      I2 = IHSTT
      KH2 = 1
      KO2 = 1
      MVL = 3
      NCPH = KHEND - IHSTT
      NCPO = KOEND - IOSTT
      MXCP = NCPO
      IF (MXCP .LT. NCPH) MXCP = NCPH
      H2IWT = 0.
      H2SWT = 0.
      O2IWT = 0.
      O2SWT = 0.
C
      DO 770 I=1,MXCP
      GO TO (700,730),KO2
C                         SET UP OXIDYZER SIDE
  700 I1 = I1 + 1
      IF (I1 .LE. KOEND) GO TO 710
      KO2 = 2
      MVL = MVL - 1
      GO TO 730
  710 CALL GETCON (I1)
      IF (CFUNCT .EQ. 1) GO TO 700
```

```
      KONAM = FNAME(CFUNCT)
      O2IWT = O2IWT + WI(I1)
      O2SWT = O2SWT + WEIGHT(I1)
C
  730 GO TO (740,760),KH2
C                        SET UP FUEL SIDE
  740 I2 = I2 + 1
      IF (I2 .LE. KHEND) GO TO 750
      KH2 = 2
      MVL = MVL - 2
      GO TO 760
  750 CALL GETCON (I2)
      IF (CFUNCT .EQ. 1) GO TO 740
      KHNAM = FNAME(CFUNCT)
      H2IWT = H2IWT + WI(I2)
      H2SWT = H2SWT + WEIGHT(I2)
C
  760 IF (MVL .EQ. 0) GO TO 780
      IF (PAGE( 1))WRITE (IOT,6110)
C                        PRINT A LINE
      CALL OUTPW (MVL,KONAM,CODE(I1),WEIGHT(I1),WI(I1),
     1                      KHNAM,CODE(I2),WEIGHT(I2),WI(I2))
  770 CONTINUE
C
  780 CONTINUE
      JP = PAGE ( 9)
      CALL SPACE
      WRITE (IOT,6112)
      TTLSWT = ENGWT + H2SWT + H2IWT + O2SWT + O2IWT
      CALL SPACE
C                        PRINT SYSTEM WT. TOTALS
      CALL OUTPF1 (4,LCNF1,ENGWT)
      DO 790 I=1,5
      CALL OUTPF1 (4,LCNF1(1,I+1),WTOFSY(I))
  790 CONTINUE
      RETURN
C
      ENTRY OTPFLT
C                        OUTPUT FLOWRATE DATA (FOR TURB.GG)
      IF (PAGE( 0)) WRITE (IOT,6120)
      JP = PAGE (15)
      CALL SPACE
      JKM = 0
      I1 = 1
      I2 = 3
  800 CALL OUTPA (1,6H      ,3,KFLUID,KFLUID(1,2))
  810 CALL SPACE
C
      DO 820 I=I1,I2
  820 CALL OUTPF (3,LFRT(1,I),3,EQRT(I,1),EQRT(I,2))
      JKM = JKM + 1
      GO TO (830,840,840),JKM
C
      ENTRY OTPFLX
C                        (FOR HEX GG FLOWRATES)
      JP = PAGE (11)
      CALL SPACE
      WRITE (IOT,6120)
```

```
      JKM = 2
C
  830 CALL SPACE
      I1 = 4
      I2 = 6
      GO TO (810,800),JKM
  840 CALL SPACE
      CALL OUTPF (3,LFRT(1,7),3,WDOTT,WDOTT(2))
      RETURN
C
      ENTRY OPAPUF (NCY)
C                       OUTPUT APU FLOWRATE DATA
      IF (PAGE( 0)) WRITE (IOT,6132) FMR
      WRITE (IOT,6130)
      JP = PAGE (6+NCY)
      DO 900 I=1,NCY
  900 WRITE (IOT,6140) I,PCTHP(I),WD(I),TE(I)
      IF (PAGE ( 9)) WRITE (IOT,6150)
      CALL SPACE
      CALL SPACE
      DO 910 I=1,5
  910 CALL OUTPF1 (4,LAPU1(1,I),EQAP1(I))
      CALL OUTPF1 (4,LAPU1(1,6),WDOTI    )
      CALL OUTPF1 (4,LAPU1(1,7),WDOTI(2))
      RETURN
C
      ENTRY OAPUSP (NCY)
C                       OUTPUT APU-SUPERCRITICAL DATA
      IF (PAGE( 0)) WRITE (IOT,6160) FMR
      JP = PAGE (9+NCY)
C
      WRITE (IOT,6162)
      DO 1000 I=1,NCY
 1000 WRITE (IOT,6161) I,Q1ODOT(I),Q1HDOT(I),WDD(I),WDA(I),
     1                          Q2ODOT(I),Q2HDOT(I),WDE(I),WDB(I)
C
      IF (PAGE (7+NCY)) WRITE (IOT,6160) FMR
      WRITE (IOT,6164)
      DO 1010 I=1,NCY
 1010 WRITE (IOT,6161) I,Q3ODOT(I),Q3HDOT(I),WDF(I),WDC(I),
     1                          WTO(I),WTH(I),DWDB(I)
C
      IF (PAGE (7+NCY)) WRITE (IOT,6160) FMR
      WRITE (IOT,6166)
      DO 1020 I=1,NCY
 1020 WRITE (IOT,6161) I,WGOC(I),WGHC(I),TTO(I),TTH(I),PCO2WD(I),
     1                          PCH2WD(I),DQODWO(I),DQODWH(I)
C
      IF (PAGE (7+NCY)) WRITE (IOT,6160) FMR
      WRITE (IOT,6168)
      DO 1030 I=1,NCY
 1030 WRITE (IOT,6161) I,PHIO2(I),PHIH2(I),Q2ODTC(I),Q2HDTC(I),
     1                          WDEC(I),WDBC(I)
C
      IF (PAGE (7+NCY)) WRITE (IOT,6160) FMR
      WRITE (IOT,6170)
      DO 1040 I=1,NCY
 1040 WRITE (IOT,6171) I,Q3ODTC(I),Q3HDTC(I),WDFC(I),WDCC(I)
```

```
C
      IF (PAGE( 0)) WRITE (IOT,6160) FMR
      JP = PAGE (18)
      CALL SPACE
      CALL OUTPA (1,6H      ,3,JFLUID,JFLUID(1,2))
      CALL SPACE
      DO 1050 I=1,11
 1050 CALL OUTPF (4,LAPU2(1,I),3,EQAP2(I,1),EQAP2(I,2))
      RETURN
C
      ENTRY OAPUSB (NCY)
C                  OUTPUT APU - SUBCRITICAL DATA
      IF (PAGE( 0)) WRITE (IOT,6172) FMR
      JP = PAGE (9+NCY)
C
      WRITE (IOT,6174)
      DO 1100 I=1,NCY
 1100 WRITE (IOT,6161) I,Q6ODOT(I),Q4HDOT(I),WDJ(I),WDG(I),
     1                    Q7ODOT(I),Q5HDOT(I),WGGO(I),WGGH(I)
C
      IF (PAGE( 0)) WRITE (IOT,6172) FMR
      JP = PAGE (18)
      CALL SPACE
      CALL OUTPA (1,6H      ,3,JFLUID,JFLUID(1,2))
      CALL SPACE
C
      DO 1110 I=1,11
      NVL = 3
      IF (I .EQ. 7) NVL = 2
 1110 CALL OUTPF (4,LAPU3(1,I),NVL,EQAP3(I,1),EQAP3(I,2))
      RETURN
C
      END
```

********* SUBRØUTINE ØUTPUT

```
C
      SUBROUTINE OUTPF (NTL,ITITL,NVL,VAR1,VAR2)
C
      DIMENSION ITITL(6),IVA1(2),IVA2(2)
C
 6001 FORMAT (29X,6A6)
 6002 FORMAT ('+'64X,E15.6)
 6003 FORMAT ('+'84X,E15.6)
 6004 FORMAT ('+'64X,I11)
 6005 FORMAT ('+'84X,I11)
 6006 FORMAT ('+'68X,2A6)
 6007 FORMAT ('+'88X,2A6)
 6008 FORMAT (' ')
 6009 FORMAT (T13,A6,T27,A6,2F14.3,T66,'.')
 6010 FORMAT (T66,'.'T73,A6,T87,A6,2F14.3)
 6011 FORMAT (T13,A6,T27,A6,2F14.3,T66,'.'T73,A6,T87,A6,2F14.3)
C
      WRITE (IOU,6001) (ITITL(I),I=1,NTL)
      GO TO (110,120,110),NVL
  110 WRITE (IOU,6002) VAR1
      IF (NVL .EQ. 1) RETURN
  120 WRITE (IOU,6003) VAR2
      RETURN
C
      ENTRY OUTPF1 (NTL,ITITL,VAR1)
C
      WRITE (IOU,6001) (ITITL(I),I=1,NTL)
      WRITE (IOU,6002) VAR1
      RETURN
C
      ENTRY OUTPI (NTL,ITITL,NVL,IVR1,IVR2)
C
      WRITE (IOU,6001) (ITITL(I),I=1,NTL)
      GO TO (210,220,210),NVL
  210 WRITE (IOU,6004) IVR1
      IF (NVL .EQ. 1) RETURN
  220 WRITE (IOU,6005) IVR2
      RETURN
C
      ENTRY OUTPA (NTL,ITITL,NVL,IVA1,IVA2)
C
      WRITE (IOU,6001) (ITITL(I),I=1,NTL)
      GO TO (310,320,310),NVL
  310 WRITE (IOU,6006) IVA1
      IF (NVL .EQ. 1) RETURN
  320 WRITE (IOU,6007) IVA2
      RETURN
C
      ENTRY OUTPW (NVL,JVA1,JVA2,VAR1,VAR3,JVA3,JVA4,VAR2,VAR4)
C
      GO TO (410,420,430),NVL
  410 WRITE (IOU,6009) JVA1,JVA2,VAR1,VAR3
      GO TO 450
  420 WRITE (IOU,6010) JVA3,JVA4,VAR2,VAR4
      GO TO 450
  430 WRITE (IOU,6011) JVA1,JVA2,VAR1,VAR3,JVA3,JVA4,VAR2,VAR4
  450 RETURN
```

```
********    OUTPUT    ********

      C
            ENTRY SPACE
      C
            WRITE (IOU,6008)
            RETURN
      C
            ENTRY OTUNIT (IONO)
      C
            IOU = IONO
            RETURN
            END
```

******** FUNCTIØN ØXENTH

```
      FUNCTION OXENTH(PRES,TEMP)
C
C     ROUTINE CALLS STEWARTS COMBINED O2-N2 PROPERTIES PROGRAM
C
      CALL ONPROP(TEMP,PRES,DX,ENTHI,SX,UX,ZX,1)
      OXENTH = ENTHI
      RETURN
      END
```

```
C                  * * * * * * * * * * * * * * * * * * * * * * * * *            PAGE 0100001
C                  * ROUTINE NAME - PAGE HEADING SUBFUNCTION.        *          PAGE 0200002
C                  * ROUTINE LANG - FORTRAN IV UNIVAC 1107/1108      *          PAGE 0300003
C                  * PROGRAMMER   - R. BOLLINGER 5432 104 22898      *          PAGE 0400004
C                  * DATE CODED   - 03/10/66                         *          PAGE 0500005
C                  * * * * * * * * * * * * * * * * * * * * * * * * *            PAGE 0600006
C                                                                               PAGE 0700007
      FUNCTION PAGE(NLINES)                                                     PAGE 0800008
C                                                                               PAGE 0900009
C                  ***** EXPLANATION OF CALLING SEQUENCE *****                  PAGE 1000010
C                                                                               PAGE 1100011
C        INPUT TO PAGE                                                          PAGE 1200012
C                                                                               PAGE 1300013
C              NLINES - NUMBER OF LINES                                         PAGE 1400014
C                       IF NLINES = -1, THE ROUTINE WILL INITIALIZE             PAGE 1500015
C                                       THE ROUTINE FOR A NEW CASE,             PAGE 1600016
C                                       PAGE EJECT, AND OUTPUT THE              PAGE 1700017
C                                       PAGE HEADING.                           PAGE 1800018
C                                    0, THE ROUTINE WILL PAGE EJECT,            PAGE 1900019
C                                       AND OUTPUT THE PAGE HEADING.            PAGE 2000020
C                                    +, THE ROUTINE WILL ADD NLINES             PAGE 2100021
C                                       TO LINECT AND TEST TO SEE IF            PAGE 2200022
C                                       LINECT IS GREATER THAN MAXLIN.          PAGE 2300023
C                                       IF IT IS, THE ROUTINE WILL              PAGE 2400024
C                                       PAGE EJECT AND OUTPUT THE PAGE          PAGE 2500025
C                                       HEADING.                                PAGE 2600026
C                                                                               PAGE 2700027
C        OUTPUT FROM PAGE                                                       PAGE 2800028
C                                                                               PAGE 2900029
C              PAGE - PAGE EJECT OR NO PAGE EJECT                               PAGE 3000030
C                     IF PAGE = TRUE , THE ROUTINE HAS PAGE EJECTED.            PAGE 3100031
C                                FALSE, THE ROUTINE HAS NOT PAGE                PAGE 3200032
C                                       EJECTED.                                PAGE 3300033
C
      LOGICAL PAGE
C                                                                               PAGE 3400036
      INCLUDE CPAGE
C                                                                               PAGE 4100038
      PAGE = .FALSE.                                                            PAGE 4200039
      IF(NLINES)10,30,20                                                        PAGE 4300040
C                                                                               PAGE 4400041
   10 NCASE = NCASE + 1                                                         PAGE 4500042
      CALL DATE(9,DOR)                                                          PAGE 4800043
      GO TO 40                                                                  PAGE 4900044
C                                                                               PAGE 5000045
   20 LINECT = LINECT + NLINES                                                  PAGE 5100046
      IF(LINECT.LE.MAXLIN) RETURN                                               PAGE 5200047
   30 LINECT = NLINES                                                           PAGE 5400048
   40 NPAGE = NPAGE + 1                                                         PAGE 5410049
      CALL TOD(8,TIME)                                                          PAGE 5500050
      PAGE = .TRUE.                                                             PAGE 5700051
      WRITE(OPTLUN,6000)NAME,NPAGE,    DEPT,PTITLE,DOR,EXT,TIME,BLD,            PAGE 5800052
     1                  JNUM,NCASE,CTITLE
      RETURN                                                                    PAGE 5900054
 6000 FORMAT ('1' T28,'NAME ' 2A6, 21(' *'),' PAGE ' I5 /
     1            T28,'DEPT ' A4,9X'*   ' 6A6,' * DATE 'A6,A3 /
     2            T28,'EXT. ' A5,8X'*'39X'* TIME ' A6,A2 /
```

```
*********    PAGE    *********

      3           T28,'BLD. ' A3,10X'*'17XA6,16X '* CASE 'I5 /
      4           T28, 39('* ') / T30, 12A6 )
C                                                                  PAGE 6000060
      END                                                          PAGE 6500061
```

```
C
C                                 * * * * * * * * * * * * * * * * * * *
C                                 * PUMP PARAMETRIC DATA ROUTINE      *
C                                 * UNIVAC 1108          FORTRAN  V   *
C                                 * J.A.MCKAY  D1943 B-201 X45178     *
C                                 * DATE CODED      AUGUST 24,1971    *
C                                 * REVISED            MARCH  1972    *
C                                 * * * * * * * * * * * * * * * * * * *
C
C
C      ****************************************************************
C      *        * * * INPUT VARIABLES * * *
C      *
C      *   IGS    = 1 FOR OXYGEN
C      *          = 2 FOR HYDROGEN
C      *
C      *   JKM    = 1 FOR MINIMUM POWER REQUIREMENT
C      *          = 2 FOR MINIMUM WEIGHT REQUIREMENT
C      *
C      *   DLP    = REQUIRED PRESSURE RISE (PSI)
C      *
C      *   WDP    = FLOWRATE REQUIRED (LB/SEC)
C      *
C      *   RNPS   = NET POSITIVE SUCTION PRESSURE AVAILABLE (NPSPA) (PSI)
C      *
C      *   RHW    = FLUID DENSITY (LBS/CU.FT.)
C      *
C      *        * * * OUTPUT VARIABLES * * *
C      *
C      *   TOTNU  = PUMP EFFICIENCY
C      *
C      *   V      = PUMP VOLUME
C      *
C      *   E      = PUMP POWER
C      *
C      *   WT     = PUMP WEIGHT
C      *
C      *   PNSG   = PUMP SPEED
C      *
C      *   NSTG   = NUMBER OF PUMP STAGES
C      *
C      *   NPSPR  = COMPUTED NPSH REQUIRED BY PUMP
C      *
C      ****************************************************************
C
       SUBROUTINE PARPMP (IGS,JKM,DLP,WDP,RNPS,RHW,TOTNU,V,E,WT,PNSG,
      1 NSTG, NPSPR)
C
       REAL NMAX,NPSPA(2),NPSPR,NSS1,NUMX(5)
C
       INCLUDE CIOUNT
       INCLUDE CONST
       INCLUDE DUMMY
       INCLUDE SPUMP,LIST
       INCLUDE TABLOK
C
       DIMENSION DELP(2),RHO(2),S(2),WDOTP(2)
```

```
********     PARPMP    *********
          DIMENSION AE(5),AV(5),AW(5)
          DIMENSION XM(12),SWB(2)
    C
          EQUIVALENCE (SCRTCH(4),NUMX),(SCRTCH(12),DELP)
         1,(SCRTCH(14),WDOTP),(SCRTCH(16),NPSPA),(SCRTCH(18),RHO)
         2,(SCRTCH(34),AE),(SCRTCH(39),AV),(SCRTCH(44),AW)
    C
          DATA S/50000.,300000./
          DATA XM /2.5,3.5,3.5,2.25,3.25,3.25,2.0,2.75,2.75,2.0,3.0,3.0 /
          DATA SWB /50.,3./ PB /1000./
    C
          DELP(IGS) = DLP
          NPSPA(IGS) = RNPS
          RHO(IGS) = RHW
          WDOTP(IGS) = WDP
          DNS = 2.0
          SQTQ = SQRT (449. * WDOTP(IGS) / RHO(IGS))
          IF (NPSPA(IGS) .EQ. 0.) GO TO 10
    C
          NMAX = S(IGS)*(144.*NPSPA(IGS)/RHO(IGS))**(.75) / SQTQ
          IF (NMAX .LT. 100000.) GO TO 20
       10 NMAX = 100000.
       20 DO 50 I=1,5
    C
    C                I  - NUMBER OF PUMP STAGES
    C
          NSTG = I
    C
    C                CALC HEAD RISE PER STAGE
    C
          H = 144.*DELP(IGS) / (I*RHO(IGS))
    C
    C                CALC MULT. FACTOR FOR SPECIFIC SPEED
    C
          XNS = SQTQ / H**(.75)
    C
    C                LOOP TO FIND MAX PUMP HYDRAULIC EFF. (NUMX)
    C
          DINT = NMAX
          NSG = 0.
          NSS1 = 0.
          DO 40 J=1,10
          DINT = DINT / 2.
          NSS1 = NSS
          NSG = NSG + DINT - 2.*DNS/XNS
          CALL PUMPEF (2)
          SD = DBLE(NU(2)) - DBLE(NU(1))
          DINT = SIGN (DINT,SD)
          NU(1) = NU(2)
          IF (ABS(NSS1-NSS) - DNS) 42,42,40
       40 CONTINUE
    C
       42 CONTINUE
          NPSPR = 0.
          IF (NPSPA(IGS) .EQ. 0.) GO TO 60
    C
    C                CALC. REQUIRED NPSP
    C
```

```
      NPSPR = RHO(IGS) / 144.*(NSG*SQTQ / S(IGS))**(4./3.)
      IF (NPSPA(IGS) .GE. NPSPR) GO TO 60
C
C                 IF NPSP (REQ) GREATER THAN NPSP (AVIL) RECALC NU
C
      NPSPR = NPSPA(IGS)
      NSG = S(IGS) / SQTQ * (144.*NPSPR/RHO(IGS))**(3./4.)
C
C                 RECALC NSS,...,NU
C
      DNS = 0.
      CALL PUMPEF (1)
      DNS = 2.0
   60 CONTINUE
      IF (U .GT. 1700.) NU = 0.
      TOTNU = NU/(1.+.05*(NSTG-1.)/NSTG)
      ISTG = NSTG
      IF (NSTG .GT. 3) ISTG = 3
      IF (NSS .LE. 375.0) GO TO 120
C
C                 FIND DIAMETER AND LENGTH OF PUMP
C                 PUMP TYPE A OR C
C
      PDI = 1.7 * DI
      IF (DI .GT. 1.5) GO TO 100
      PLGT = XM(ISTG)*DI
      GO TO 130
  100 IF (DI .GT. 2.0) GO TO 110
      PLGT = XM(ISTG+3)*DI
      GO TO 130
  110 PLGT = XM(ISTG+6)*DI
      GO TO 130
C
C                 PUMP TYPE B
C
  120 PDI = 1.4 * DI
      PLGT = XM(ISTG+9)*DI
  130 IF (NSTG .LT. 3) GO TO 140
      PLGT = PLGT + DI*(NSTG-2) / 2.
  140 CONTINUE
C
      CALL FINTAB (NTBID(22))
      WB = MIPE (1,DI)
      FW = (WDOTP(IGS) / SWB(IGS))**(.25)
      P =  NPSPA(IGS) + DELP(IGS)
C
C                 CALCULATE PUMP WEIGHT
C
      IF (NSS .GT. 375.0) GO TO 150
C
C                 PUMP TYPE B
C
      WI = .05
      WH = .35
      WS = NSTG**(.75)
      GO TO 160
C
C                 PUMP TYPE A OR C
```

```
********     PARPMP    ********

      C
        150 WI = .1*FW
            WH = .55
            WS = 1.
        160 WI = WI*NSTG*WB
            WH = WH*WB*NSTG**(1./3.)
            WS = WS*.35*WB*P/PB*FW
            WT = WI + WH + WS
      C
      C                      CALC. PUMP VOLUME
      C
            V = .25*PI*PLGT*PDI*PDI
      C
      C                      CALC. PUMP POWER
      C
            E = WDOTP(IGS)*H*NSTG / (550.*TOTNU)
            AE(I) = E
            AV(I) = V
            AW(I) = WT
            NUMX(I) = TOTNU
      C
         50 CONTINUE
            XMN = 1.E10
            DO 230 K=1,5
            GO TO (200,210),JKM
      C
      C                      FOR MIN. POWER REQ.
      C
        200 IF (AE(K) .GE. XMN) GO TO 230
            XMN = AE(K)
            GO TO 220
      C
      C                      FOR MIN. WEIGHT REQ.
      C
        210 IF (AW(K) .GE. XMN) GO TO 230
            XMN = AW(K)
        220 NSTG = K
        230 CONTINUE
            E = AE(NSTG)
            V = AV(NSTG)
            WT = AW(NSTG)
            TOTNU = NUMX(NSTG)
            PNSG = NSG
      C
      C                      OFF-DESIGN PUMP PERFORMANCE
      C
            RETURN
            END
```

******** SUBRØUTINE PFND

```
      SUBROUTINE PFND(T,D,P)
      COMMON /RFPR/ RF(10)
      COMMON /CEOS/G(41)
      COMMON /SCRH/ B(40)
C
C.... THIS ROUTINE CALCULATES PRESSURE GIVEN TEMPERATURE AND DENSITY
C.... FROM THE EQUATION OF STATE
C
      R=RF(5)
      D2=D*D
      D3=D2*D
      D4=D3*D
      D5=D4*D
      D6=D5*D
      D7=D6*D
      D8=D7*D
      D9=D8*D
      D10=D9*D
      D11=D10*D
      D12=D11*D
      D13=D12*D
      TS  = SQRT(T)
      T2=T*T
      T3=T2*T
      T4=T3*T
      GM=G(41)
      F   = EXP(GM*D2)
      B( 1)=D2*T
      B( 2)=D2*TS
      B( 3)=D2
      B( 4)=D2/T
      B( 5)=D2/T2
      B( 6)=D3*T
      B( 7)=D3
      B( 8)=D3/T
      B( 9)=D3/T2
      B(10)=D4*T
      B(11)=D4
      B(12)=D4/T
      B(13)=D5
      B(14)=D6/T
      B(15)=D6/T2
      B(16)=D7/T
      B(17)=D8/T
      B(18)=D8/T2
      B(19)=D9/T2
      B(20)=D3*F/T2
      B(21)=D3*F/T3
      B(22)=D5*F/T2
      B(23)=D5*F/T4
      B(24)=D7*F/T2
      B(25)=D7*F/T3
      B(26)=D9*F/T2
      B(27)=D9*F/T4
      B(28)=D11*F/T2
      B(29)=D11*F/T3
      B(30)=D13*F/T2
```

```
*********     PFND     *********

          B(31)=D13*F/T3
          B(32)=D13*F/T4
          N=32
          P    = 0.0
          DO 1 I=1,N
        1 P=P+B(I)*G(I)
          P=P+R*D*T
          RETURN
          END
```

******** SUBRØUTINE PFNDB

```
      SUBROUTINE PFNDB(TB,DB,PB)
      COMMON /RFPR/ RF(10)
      WT=RF(7)
      T   = TB/1.8
      D   = DB * 453.59237E-3/(WT * 2.8316847E-2)
      CALL PFND(T,D,P)
      PB  = P * 1.01325E+5/6.8947572E+3
      RETURN
      END
```

******** SUBRØUTINE PHIB

```
      SUBROUTINE PHIB(DB,CVB,D2B,EDB)
C
C     CALCULATES THE ENERGY DERIVATIVE FROM THE EQUATION OF STATE IN
C     BRITISH UNITS
C     MUST FOLLOW CALLS OF PROPB,CPVTDB,AND DPDTB TO DEFINE DB, CVB, +
C     D2
C     INPUT AND OUTPUT PARAMETERS ARE IN BRITISH UNITS
C
      VB  = 1.0/DB
      EDB=(VB/CVB)*D2B
      RETURN
      END
```

******** SUBRØUTINE PHTHØN

```
      SUBROUTINE PHTHON(TEMP,DENS,NGAS,PHI,THETA)
C
      COMMON /METH/ M
C
C     INPUT TO THIS SUBROUTINE MUST BE IN BRITISH UNITS
C
      T = TEMP
      D = DENS
      M = 1
C
      IF(NGAS.EQ.1) KF = 1
      IF(NGAS.EQ.18) KF = 2
C
C     KF = 1  CALL IN OXYGEN PARAMETERS
C     KF = 2  CALL IN NITROGEN PARAMETERS
C
      IF(KF.EQ.1) CALL DATAO2
      IF(KF.EQ.2) CALL DATAN2
C
      D1=DPDDB(T,D)
      D2=DPDTB(T,D)
      CALL CPVTDB(T,D,CP,CV)
      CALL THETAB(D,CP,D1,D2,SHI)
      THETA = SHI
C
      CALL PHIB(D,CV,D2,ED)
      PHI = ED
C
      RETURN
      END
```

******** SUBRØUTINE PRØP

```
      SUBROUTINE PROP(T,P,D,K,H,S,U,Z)
      COMMON /CRPR/ CR(3) /METH/M
C
C.... GENERALIZED PROPERTY CALCULATOR
C
C.... ROUTINE CALCULATES PROPERTIES FOR FOLLOWING INPUT OF K
C
C     K=1   INPUT IS T + P     RETURNS D, H, S, + U
C     K=2   INPUT IS T + D     RETURNS P, H, S, + U
C     K=3   INPUT IS T         RETURNS P, D, H, S, + U FOR SATURATED VAPOR
C     K=4   INPUT IS T         RETURNS P, D, H, S, + U FOR SATURATED LIQUID
C
C.... NOTE     ALL REAL VARIABLES IN CALL STATEMENTS TO ROUTINES IN
C              THIS PACKAGE MUST BE TYPED IMPLICIT DOUBLE PRECISION
C              (A-H, O-Z)
C
C.... NOTE     THE FIRST CALL STATEMENT IN THE USER'S MAIN PROBRAM MUST
C              BE TO A DATA INITIALIZATION ROUTINE
C              EXAMPLE   CALL DATA02
C
C.... NOTE     THE METHOD OF PROPERTY CALCULATION IS DETERMINED BY THE
C              VALUE OF M CONTAINED IN COMMON BLOCK /METH/M
C
C     M=1    INDICATES PROPERTY CALCULATION TO BE CARRIED OUT BY
C            CONTINUOUS INTEGRATION OF ISOTHERMS THROUGH THE TWO PHASE
C            REGION
C     M=2    INDICATES PROPERTY CALCULATION IS INTURUPTED AT THE TWO
C            PHASE VAPOR BOUNDRY AND THE CLAPEYRON RELATION WITH THE
C            VAPOR PRESSURE EQUATION IS USED TO CALCULATE THE LATENT
C            HEAT.  INTEGRATION OF ISOTHERMS IS CONTINUED AT THE
C            SATURATED LIQUID BOUNDRY
C
      PC=CR(1)
      DC=CR(2)
      TC=CR(3)
      IF((K.GT.0).AND.(K.LT.5))GO TO 1
      WRITE(6,300)K
  300 FORMAT(' *** ERROR IN CALL PROP ***',/,
     .       '     K MUST EQUAL 1,2,3, OR 4',/,
     .       '     K = ',I10)
      RETURN
    1 IF(K.LT.3)GO TO 3
      IF(T.LE.TC)GO TO 2
      WRITE(6,301)T
  301 FORMAT(' *** ERROR IN CALL PROP ***',/,
     .       '     SATURATION PROPERTIES HAVE BEEN REQUESTED',/,
     .       '     FOR A TEMPERATURE TAHT EXCEEDS CRITICAL',/,
     .       '     T = ',G15.5)
    2 P=VPN(T)
       IF(K.EQ.3)CALL DFND(T,P,D,Z1,2)
       IF(K.EQ.4)CALL DFND(T,P,D,Z1,1)
      GO TO 4
    3 IF(K.GT.1)GO TO 7
       CALL DFND(T,P,D,Z1,0)
      IF(T.GT.TC)GO TO 5
    4 IF(D.GT.DC)GO TO 6
    5 CALL VPROP(T,P,D,3,H,S,U,Z)
```

```
      RETURN
    6 IF(M.EQ.1)GO TO 5
      CALL LPROP(T,P,D,3,H,S,U,Z)
      RETURN
    7 IF(T.GT.TC)GO TO 8
      VP=VPN(T)
       CALL DFND(T,VP,DV,Z2,2)
       CALL DFND(T,VP,DL,Z2,1)
      IF(D.GE.DL)GO TO 9
      IF(D.GT.DV)GO TO 10
    8 CALL VPROP(T,P,D,1,H,S,U,Z)
      RETURN
    9 IF(M.EQ.1)GO TO 8
      CALL LPROP(T,P,D,1,H,S,U,Z)
      RETURN
   10 VL  = 1.0/DL
      VV  = 1.0/DV
      V   = 1.0/D
      X=(V-VL)/(VV-VL)
      CALL VPROP(T,P,DV,1,HV,SV,UV,ZV)
      IF(M.EQ.2)GO TO 11
      CALL VPROP(T,P,DL,1,HL,SL,UL,ZL)
      GO TO 12
   11 CALL LPROP(T,P,DL,1,HL,SL,UL,ZL)
   12 H=HL+X*(HV-HL)
      S=SL+X*(SV-SL)
      U=UL+X*(UV-UL)
      Z = ZL
      RETURN
      END
```

********* SUBRØUTINE PRØPB

```
      SUBROUTINE PROPB(TB,PB,DB,K,HB,SB,UB,ZB)
      COMMON /RFPR/RF(10)
      WT=RF(7)
      T   = TB/1.8
      P   = PB * 6.8947572E+3/1.01325E+5
      D   = DB * 453.59237E-3/(WT * 2.8316847E-2)
      CALL PROP(T,P,D,K,H,S,U,Z)
      PB  = P * 1.01325E+5/6.8947572E+3
      DB  = D * WT * 2.8316847E-2/453.59237E-3
      HB  = H * 453.59237/(1.0543502E+3 * WT)
      UB  = U * 453.59237/(1.0543502E+3 * WT)
      SB  = S * 453.59237/(1.0543502E+3 * 1.8 * WT)
      ZB = Z
      RETURN
      END
```

******** FUNCTIØN PSATH

```
      FUNCTION PSATH(PRESS,HG,HL)
      DIMENSION R(19),TL(19),TG(19),TF(19)
      DATA R/1.022,2.0,4.0,8.0,14.0,25.0,43.0,69.0,99.0,128.0,151.0,
     1165.,176.0,182.0,185.0,186.5,187.25,187.46875,187.506/
      DATA                                                    TF/24.845,
     227.07,29.81,33.07,36.18,39.96,44.12,48.33,51.97,54.79,56.72,57.80,
     358.57,58.99,59.18,59.29,59.34,59.353,59.356/
      DATA                                             TG/60.31,65.11,70.59
     4,76.35,80.98,85.11,87.40,86.54,81.94,74.15,64.83,56.86,47.34,39.56
     5,33.46,28.34,22.31,18.66,16.55/
      DATA                                  TL/-132.8,-129.13,-124.25,-117.79
     6,-110.86,-101.3,-89.04,-74.22,-58.58,-43.43,-30.07,-20.56,-11.13,
     7-4.27,1.17,5.54,10.83,14.29,16.36/
      P=PRESS
      IF(P.LT.1.022)P=1.022
      IF(P.GE.187.506)P=187.506
      DO 104 I=2,19
      IF(P-R(I))102,101,104
  101 HL=TL(I)
      HG=TG(I)
      PSATH =TF(I)
      RETURN
  102 D=R(I)-R(I-1)
      PRR=R(I)-P
      PPR=P-R(I-1)
      HL=(TL(I)*PPR+TL(I-1)*PRR)/D
      HG=(TG(I)*PPR+TG(I-1)*PRR)/D
      PSATH =(TF(I)*PPR+TF(I-1)*PRR)/D
      RETURN
  104 CONTINUE
      RETURN
      END
```

********* PTDENS *********

```
      DATA               AD /5.033,5.112,5.183,4.816,4.910,4.994,4.558,4.
     1676,4.764,4.248,4.405,4.533,3.880,4.094,4.246,3.442,3.743,3.958,2.
     2953,3.361,3.633,2.481,2.975,3.307,2.093,2.616,3.002,1.804,2.309,2.
     3698,1.588,2.059,2.462,1.423,1.856,2.226,1.293,1.691,2.056,1.187,1.
     4555,1.886,1.100,1.441,1.754,1.026,1.344,1.639,.9627,1.262,1.546,.9
     5076,1.189,1.453,  4.742,4.853,4.946,5.031,4.430,4.584,4.709,4.816
     6,4.024,4.279,4.417,4.558,3.165,3.778,4.050,4.248,1.430,2.988,3.595
     7,3.880,   -.0233,.5581,1.189,1.804,-.015,.5030,1.05,1.588,-.0099,.
     84591,.945,1.423,-.0065,.4231,.8618,1.293,-.0042,.3928,.7943,1.187,
     9-.0025,.3669,.7379,1.1,-.0015,.3445,.6898,1.026,-.0007,.3249/
      DATA               AE /.6483,.9627,-.0001,.3075,.6120,.9076/
      DATA               AF /-.07704,.02568,.1307,.2384,-.06786,.02262,.1
     1144,.2079,-.06066,.02022,.1018,.1844,-.05484,.01828,.09179,.1659,-
     2.05004,.01668,.0836,.1508,-.04602,.01534,.07677,.1383,   3.887,3.9
     386,4.105,3.447,3.646,3.792,2.910,3.226,3.442,2.323,2.760,3.051,1.8
     457,2.317,2.670,1.549,1.963,2.326,1.34,1.701,2.035,1.189,1.497,1.80
     54,   .7859,1.309,1.857,.6864,1.106,1.549,.6141,.9689,1.34,.5581,.8
     6735,1.189,   .1675,.3524,.5577,.7859,.1530,.3178,.4955,.6864,.1409,
     7.2902,.4479,.6141,.1307,.2674,.4127,.5581,  -.03228,.03224,.09858,
     8.1675,-.029742,.02971,.09043,.153,-.027561,.02754,.08359,.1409,-.0
     925680,.02568,.07819,.1307/
      DATA               AG /1.226,1.807,2.329,2.674,2.91,1.052,1.481,1.9
     141,2.326,2.616,.9378,1.279,1.656,2.019,2.323,.8526,1.141,1.456,1.7
     274,2.089,.7859,1.047,1.309,1.583,1.857,  .09975,.2084,.3279,.461,.6
     3115,.7847,.9864,1.226,.09478,  .1962,.3063,.4264,.5585,.7052,.8693,
     41.052,.08981,.1855,.2878,.3977,.5164,.6451,.7851,.9388,.08590,.176
     5,.2717,.3734,.4817,.5972,.7207,.8526,.08199,.1675,.2599,.3524,.455
     60,.5577,.6718,.7859,  -.019364,.01932,.05888,.09975,-.017558,.0175
     73,.05323,.08981,-.01615,.01611,.04882,.08199,   3.304,3.422,3.541,
     83.659,3.777,3.044,3.290,3.450,3.572,3.667,2.740,3.102,3.304,3.449,
     93.557,2.322,2.880,3.141,3.315,3.447,1.861,2.621,2.959,3.170,3.313/
      DATA               AH /1.551,2.329,2.760,3.014,3.179,1.358,2.045,
     12.546,2.848,3.044,1.226,1.807,2.329,2.674,2.910/
      DATA               AI /.6295,2.42,2.726,2.849,2.937,3.005,3.061,3.1
     11,3.153,3.191,3.227,3.259,.9338,1.076,1.265,1.599,2.202,2.506,2.65
     29,2.768,2.85,2.918,2.976,3.026,.8297,.9234,1.03,1.159,1.322,1.542,
     31.835,2.136,2.356,2.509,2.618,2.709,.7590,.8323,.9121,1.001,1.103,
     41.220,1.359,1.523,1.713,1.916,2.106,2.267,.7053,.7670,.8326,.9039,
     5.9807,1.066,1.160,1.265,1.383,1.513,1.655,1.803,.6621,.7161,.7728,
     6.8331,.8976,.9656,1.040,1.120,1.206,1.299,1.400,1.507,.6259,.6745,
     7.7250,.7780,.8337,.8927,.9537,1.019,1.089,1.163,1.242,1.325,.5941,
     8.6412,.6883,.7355,.7826,.8372,.8921,.9469,1.005,1.070,1.135,1.200/
      DATA              AJ/-.000743,.04849,.09854,.0,.0,-.000787,.04023,.
     108466,.1315,.1915,-.000437,.03393,.07032,.1097,.1534,-.000257,.029
     241,.06027,.09278,.1275,-.000177,.02599,.05289,.0808,.1098,-.000122
     3,.0233,.0472,.07177,.09714,-.000083,.02112,.04266,.06465,.08719,-.
     4000065,.01932,.03810,.05888,.07931,  0.,.01172,.02363,-.000111,.00
     59736,.0198,-.000059,.008154,.01648,-.000036,.007021,.01414,-.00002
     64,.006166,.0124,-.000015,.005498,.01104,-.000011,.004962,.009954,-
     7.000009,.004521,.009063,-.000007,.004153,.008321,-.000022,.003846,
     8.007715/
      DATA AK/.1664,.248,0.,0.,0.,0.,.1469,.2418,.3397,0.,0.,0.,.1315,
     1.2117,.3091,.4233,.5318,0.,.1195,.1891,.2684,.3626,.4811,.6115,
     2.1098,.1717,.2400,.3166,.4051,.5124,.1017,.1578,.2183,.2841,.3568,
     3.4387,.09486,.1463,.2009,.2592,.322,.3903,.08892,.1377,.1865,.2407
     4,.295,.3565,.5712,.727,.9725,0.,0.,0.,.5124,.6559,.8514,1.116,0.,
     50.,.4755,.5827,.7329,.9604,1.252,1.718,.4387,.5333,.6477,.7994,1.0
     668,1.534,.4145,.4956,.5914,.7060,.8535,1.082,.3903,.4653,.5491,.64
```

```
********     PTDENS     ********

          747,.7575,.8985,.3734,.4399,.5151,.5986,.6928,.8020,.3565,.4181,.48
          898,.5616,.6487,.7358,.08892,.1865,.295,.4181,.5616,.7358,.08373,.1
          9743,.2731,.3822,.5044,.6439,.07931,.1649,.2562,.3545,.4610,.5814/
           P=PRES
           IF(P.LT.1.0) P=1.0
           T=TEMP
           IF(T.LT.180.0) GO TO 8
           IF(T.GE.1300.0) GO TO 4
           IF(T.GE.490.0) GO TO 2
           IF(P.GE.800.) GO TO 1
           N=1
           GO TO 33
         1 N=2
           GO TO 33
         2 IF(P.GE.300.0) GO TO 3
           N=3
           GO TO 33
         3 N=4
           GO TO 33
         4 IF(T.GE.2500.0) GO TO 6
           IF(P.GE.100.0) GO TO 5
           N=5
           GO TO 33
         5 N=6
           GO TO 33
         6 IF(T.GE.5000.0) T=4999.99999
           IF(P.GE.10.0) GO TO 7
           N=7
           GO TO 33
         7 N=8
           GO TO 33
         8 TZ=24.84+0.00317*P
           IF(T.LT.TZ) T=TZ
           IF(P.LT.881.76) GO TO 11
           IF(P.LT.2645.28) GO TO 9
           N=9
           GO TO 33
         9 IF(P.LT.1469.6) GO TO 10
           N=10
           GO TO 33
        10 N=11
           GO TO 33
        11 IF(T.GE.59.4) GO TO 14
           N=12
           IF(P.GE.187.6385) GO TO 33
           DO 12 I=2,20
           IF(P-PS(I))13,13,12
        12 CONTINUE
           I=20
        13 TM=TS(I-1)+(TS(I)-TS(I-1))*(P-PS(I-1))/(PS(I)-PS(I-1))
           IF(T.GE.TM) GO TO 24
           GO TO 33
        14 IF(T.LT.108.0) GO TO 16
           IF(P.LT.132.264) GO TO 15
           N=13
           GO TO 33
        15 N=14
           GO TO 33
```

```
*********     PTDENS    *********

      16 IF(P.LT.587.84) GO TO 17
         N=15
         GO TO 33
      17 IF(T.LT.72.0) GO TO 23
         IF(T.LT.86.4) GO TO 20
         IF(P.LT.293.92) GO TO 18
         N=16
         GO TO 33
      18 IF(P.LT.73.48) GO TO 19
         N=17
         GO TO 33
      19 N=18
         GO TO 33
      20 IF(P.LT.293.92) GO TO 21
         N=19
         GO TO 33
      21 IF(P.LT.36.74) GO TO 22
         N=20
         GO TO 33
      22 N=21
         GO TO 33
      23 IF(P.LT.293.92) GO TO 24
         N=22
         GO TO 33
      24 IF(P.LT.180.0) GO TO 25
         N=23
         GO TO 33
      25 IF(P.GE.29.0) GO TO 27
         IF(P.LT.2.9392) GO TO 26
         N=24
         GO TO 33
      26 N=25
         GO TO 33
      27 IF(T.GE.64.8) GO TO 29
         IF(P.GE.102.0) GO TO 28
         N=26
         GO TO 33
      28 N=27
         GO TO 33
      29 N=28
      33 FP=(P-BP(N))/DP(N)
         IP=FP
         IF(IP.GT.MX(N)) IP=MX(N)
         FI=IP
         F=FP-FI
         FP=1.0-F
         FT=(T-BT(N))/DT(N)
         IT=FT
         FI=IT
         FF=FT-FI
         FT=1.0-FF
         I=IT*JP(N)+IP+LOC(N)
         J=I+JP(N)
         PTDENS=FP*FT*R(I)+F*FT*R(I+1)+FP*FF*R(J)+F*FF*R(J+1)
         RETURN
         END
```

******** FUNCTIØN PTHEAT

```
      FUNCTION PTHEAT(PRES,TEMP,KTRANS)
      COMMON/SPHEAT/CP(823),CV(823)
      DIMENSION LOC(23),JP(19),MX(19),BP(19),DP(19),BT(23),DT(19),PS(12)
     1,TS(12),TL(10)
      DATA PS/1.022,2.,4.,8.,14.,25.,43.,69.,99.,128.,151.,160./
      DATA TS/24.845,27.07,29.81,33.07,36.18,39.96,44.12,48.33,51.97,
     154.79,56.72,57.46/
      DATA LOC/1,50,71,113,133,141,170,165,228,264,348,372,426,492,528,
     1584,626,638,734,654,682,694,718/
      DATA JP/7,3,6,5,2,4,9,9,4,7,4,9,6,6,7,7,2,2,5/
      DATA MX/5,1,4,3,0,2,2,4,2,5,2,7,4,4,5,5,0,0,3/
      DATA BP/0.,20.,0.,1000.,100.,-1000.,1469.6,0.,1469.6,587.84,587.84
     1,0.,100.,100.,40.,40.,0.,0.,1./
      DATA BT/2600.,2600.,2600.,2600.,800.,300.,120.,120.,25.2,27.,72.,
     181.,25.,56.,26.,41.,25.,25.,2600.,5000.,5000.,5000.,5000./
      DATA DP/5.,40.,200.,1000.,4900.,2000.,1175.68,293.92,1175.68,
     1146.96,146.96,73.48,100.,100.,20.,20.,40.,40.,1./
      DATA DT/400.,400.,400.,800.,600.,100.,30.,30.,12.6,9.,3.6,9.,5.,
     15.,4.,8.,4.,8.,200./
      DATA TL/24.846,27.175,29.310,31.299,33.176,34.962,36.672,38.317,
     139.904,41.456/
    1 P=PRES
      IF(P.LT.1.0) P=1.0
      T=TEMP
      KTR=KTRANS
      IF(T.LT.126.) GO TO 9
      IF(T.LT.2600.) GO TO 5
      IF(T.GE.6000.) T=5999.99999
      IF(P.GE.100.) GO TO 3
      IF(P.GE.30.) GO TO 2
      IF(P.GE.5.)GO TO 20
      N=19
      N1=19
      GO TO 33
   20 N=1
      N1=20
      GO TO 33
    2 N=2
      N1=21
      GO TO 33
    3 IF(P.GE.1000.) GO TO 4
      N=3
      N1=22
      GO TO 33
    4 N=4
      N1=23
      GO TO 33
    5 IF(T.LT.300.)GO TO 7
      IF(T.LT.800.)GO TO 6
      N=5
      GO TO 33
    6 N=6
      GO TO 33
    7 IF(P.LT.1469.6) GO TO 8
      N=7
      GO TO 33
    8 N=8
```

```
      GO TO 33
    9 IF(P.LT.587.84) GO TO 12
      IF(P.LT.1469.6) 30 TO 10
      N=9
      GO TO 30
   10 IF(P.LT.1028.72.AND.T.GE.72.0.AND.T.LT.90.0) GO TO 11
      N=10
      GO TO 30
   11 N=11
      GO TO 33
   12 IF(T.LT.81.) GO TO 13
      N=12
      GO TO 33
   13 IF(P.LT.160.) GO TO 15
      TM=((.86867647E-7*P-.12613701E-3)*P+.10353383)*P+43.8056878
      IF(T.GT.TM) GO TO 14
      N=13
      GO TO 30
   14 N=14
      GO TO 33
   15 DO 16 I=2,12
      IF(P-PS(I))17,17,16
   16 CONTINUE
      I=12
   17 TM=TS(I-1)+(TS(I)-TS(I-1))*(P-PS(I-1))/(PS(I)-PS(I-1))
      IF(T.GE.TM) GO TO 18
      N=15
      IF(P.LT.40.) N=17
      GO TO 30
   18 N=16
      IF(P.LT.40.) N=18
      GO TO 33
   30 F=P/587.84
      I=F
      IF(I.GT.8) I=8
      FI=I
      F=F-FI
      TQ=(1.0-F)*TL(I+1)+F*TL(I+2)
      IF(T.LT.TQ) T=TQ
   33 IF(T.LE.5000.)N1=N
      FP=(P-BP(N))/DP(N)
      IP=FP
      IF(IP.GT.MX(N)) IP=MX(N)
      FI=IP
      F=FP-FI
      FP=1.0-F
      FT=(T-BT(N1))/DT(N)
      IT=FT
      FI=IT
      FF=FT-FI
      FT=1.0-FF
      I=IT*JP(N)+IP+LOC(N1)
      J=I+JP(N)
      IF(KTR.EQ.2) GOTO 37
      CTCP=FP*FT*CP(I)+F*FT*CP(I+1)+FP*FF*CP(J)+F*FF*CP(J+1)
      IF(N.LT.13.OR.N.GE.17) GO TO 36
      IF(N.LT.15) GO TO 35
      CTCP=CTCP/(187.506-P+ABS (T-TM)*28.13)
```

******** PTHEAT ********

```
      GO TO 36
   35 CTCP=CTCP/(ABS (T-TM)/1.8+ABS (P-187.506)*.008008982)
   36 IF(KTR.GE.2) GO TO 37
      PTHEAT=CTCP
      RETURN
   37 PTHEAT=FP*FT*CV(I)+F*FT*CV(I+1)+FP*FF*CV(J)+F*FF*CV(J+1)
      IF(KTR.LT.3) RETURN
      PTHEAT=CTCP/PTHEAT
      RETURN
      END
```

******** SUBRØUTINE PVAPØR

```
      SUBROUTINE PVAPOR(T,I,P)
      GO TO (1,2,3,2,5,6,7,6,7,10,3,12,13,14,15,16),I
    1 P=EXP(12.04-1519./T)
      RETURN
    2 P=10.**(2.9303-79.821/T+.011628*T)
      RETURN
    3 P = EXP(11.63-1374./T)
      IF(P.GT.200.) P = EXP(13.43-1763./T)
      RETURN
    5 P = .825*EXP(11.63-1374./T)+.175*EXP(12.04-1519./T)
      IF(P.GT.200.) P = .825*EXP(13.43-1763./T)+.175*EXP(12.04-1519./T)
      RETURN
    6 P = EXP(11.83-1839./T)
      RETURN
    7 P = 10.**(5.73-1050./T)
      RETURN
   10 P = EXP(12.3579-3168.7/T)
      RETURN
   12 P = EXP(14.45-5090./T)
      RETURN
   13 P = EXP(16.54098-7.3483*(1000./T))
      RETURN
   14 P = EXP(13.4055-6.65*(1000./T))
      RETURN
   15 PLOGMM = 7.4837-1.8*1197./T
      P = .01934*(10.**PLOGMM)
      RETURN
   16 PLOGMM = 8.2875-1.8*1996./T
      P = .01934*(10.**PLOGMM)
      RETURN
      END
```

******** SUBRØUTINE RHØLIQ

```
      SUBROUTINE RHOLIQ(T,I,R)
      GO TO (1,2,3,2,5,6,7,6,7,10,3,12,13,14,15,16),I
    1 R = 62.43*(1.6983-.003418*T)
      RETURN
    2 R=18.25132T-1.4162715*T+.058929936*T*T-.00121374846*T*T*T+
     1.121340376E-4*T**4-.48344473E-7*T**5
      RETURN
    3 R = 135.8864-.272046*T
      RETURN
    5 R = 129.5-.25*T
      RETURN
    6 R = 38.38-.0589*T
      RETURN
    7 R = 134.45-.169*T
      RETURN
   10 R = 62.43*(.76188-.000978*T)
      RETURN
   12 R = 62.4153-.046113*T
      RETURN
   13 R = 131.336-.07795*T
      RETURN
   14 R = 73.374-.032187*T
      RETURN
   15 R = 62.43*(2.805909-.001944*T)
      RETURN
   16 R = 62.43*(1.25214-.000453*T)
      RETURN
      END
```

********* SUBRØUTINE SPHSEG

```
C
      SUBROUTINE SPHSEG (PVOL,RAD,H)
C
      INCLUDE CONST,LIST
C
      DIMENSION Y(3)
C
C                   CALC. VOL. OF TOTAL HEMISPHERE
      TVOL = PI203 * RAD**3
      GO TO 70
C                   ENTRY FOR ELLIPTICAL SPHEROID
C                   RAD ALONG AXIS OF ROTATION
      ENTRY ELIPSG (PVOL,RAD,RPD,H)
C
      TVOL = PI203*RPD*RPD*RAD
   70 IF (PVOL .NE. 0.) GO TO 80
      H = 0.
      RETURN
   80 CONTINUE
      XM = PVOL / TVOL
      IF (XM .GT. 0.) GO TO 90
      WRITE (6,1002) XM
      RETURN
   90 CONTINUE
      PHI3 = ACOS (1.0-XM) / 3.0
      DO 100 I=1,3
      XI = I - 1
      Y(I) = RAD*(1.0 + 2.0*COS (PHI3 + XI*PI203))
  100 CONTINUE
      DO 200 I=1,3
      K = I
      IF (Y(I) .GT. 0. .AND. Y(I) .LT. RAD) GO TO 120
  200 CONTINUE
      WRITE (6,1000) Y
      RETURN
  120 H = Y(K)
      RETURN
C
C                   FIND HEAD IN CYLINDER
C
      ENTRY CYLHED (PVOL,RAD,H)
      H = PVOL / (PI*RAD*RAD)
      RETURN
C
C                   CALC. HEAD IN FRUSTRUM OF CONE
C
      ENTRY FRHEAD (PVOL,RTOP,RBOT,HGT,H)
      RBMT = RBOT - RTOP
      VD = (PI/3.0)*HGT/RBMT*RBOT*RBOT*RBOT - PVOL
      H = (PI*RBOT*HGT-(3.0*(PI*HGT)**2*RBMT*VD)**(1./3.)) /
     1  (PI*RBMT)
      RETURN
C
C                   RAD ALONG AXIS OF ROTATION
      ENTRY CYMSPH (PVOL,RAD,RPD,H)
C
C                   TO CALC. HEAD IN A VOLUME BETWEEN A CYLINDER AND
```

```
C                      SPHEROID
      D = 3.0*RAD*RAD*PVOL / (PI*RPD*RPD)
      H = RAD - (RAD**3 - D)**(1./3.)
      RETURN
C
 1000 FORMAT ('0' 10X 'COULD NOT FIND H FOR HEMISPHERE' 3F15.4/)
 1002 FORMAT ('0' 10X 'ERROR INPUT TO SPHSEG' F15.7)
C
      END
```

******** BLØCK DATA - SPHTDA

```
      BLOCK DATA                                                        1429 0001
      COMMON/SPHEAT/AA(111),AB(111),AC(111),AD(111),AE(116),AF(112),    1430 0002
     1AG(111),AH(40),                                                   1431 0003
     2         AI(110),AJ(111),AK(111),AL(111),AM(116),AN(112),AO(111)  1432 0004
     3,AP(41)                                                           1433 0005
      DATAAA/3.804,3.796,3.794,3.794,3.793,3.793,3.793,4.025,3.951,3.937 1434 0006
     1,3.931,3.927,3.924,3.922,4.655,4.259,4.184,4.15,4.13,4.117,4.107,6 1435 0007
     2.495,5.032,4.755,4.632,4.559,4.509,4.472,10.98,6.839,6.054,5.706,5 1436 0008
     3.498,5.356,5.252,20.06,10.48,8.646,7.834,7.35,7.02,6.776,35.35,16. 1437 0009
     483,13.18,11.55,10.58,9.919,9.429,3.793,3.792,3.792,3.924,3.917,3.9 1438 0010
     513,4.117,4.076,4.059,4.51,4.359,4.298,5.36,4.931,4.757,7.028,6.027 1439 0011
     6,5.62,9.936,7.923,7.103,3.793,3.791,3.791,3.791,3.791,3.79,3.916,3 1440 0012
     7.91,3.908,3.907,3.906,3.905,4.075,4.043,4.031,4.025,4.022,4.02,4.3 1441 0013
     86,4.236,4.192,4.172,4.161,4.183,4.933,4.581,4.457,4.401,4.369,4.34 1442 0014
     96,6.03,5.21,4.92,4.791,4.715,4.759,7.93,6.276,5.692,5.433,5.279/   1443 0015
      DATAAB/5.173,3.79,3.79,3.789,3.789,3.789,4.02,4.014,4.011,4.01,4.0 1444 0016
     108,4.346,4.29,4.265,4.25,4.24,5.173,4.912,4.796,4.726,4.679,3.47,3 1445 0017
     2.511,3.504,3.507,3.624,3.622,3.792,3.789,3.744,4.106,4.297,4.346,3 1446 0018
     3.658,3.842,3.961,4.015,3.537,3.637,3.708,3.749,3.479,3.535,3.579,3 1447 0019
     4.608,3.461,3.495,3.523,3.542,3.461,3.478,3.494,3.511,2.586,2.981,3 1448 0020
     5.396,3.693,3.806,3.785,3.434,3.187,3.036,2.861,3.089,3.309,3.498,3 1449 0021
     6.628,3.693,3.64,3.501,3.379,3.2,3.342,3.474,3.584,3.673,3.737,3.8, 1450 0022
     73.744,3.681,3.519,3.633,3.738,3.826,3.903,3.957,4.03,4.041,4.015,3 1451 0023
     8.744,3.838,3.921,3.994,4.06,4.111,4.223,4.248,4.243,3.877,3.951,4. 1452 0024
     9018,4.079,4.133,4.177,4.292,4.333,4.343,3.926,3.98,4.033,4.087/    1453 0025
      DATAAC/4.125,4.152,4.263,4.317,4.346,1.334,1.264,1.217,1.2,1.884,1 1454 0026
     1.742,1.667,1.587,2.387,2.151,2.025,1.947,2.871,2.506,2.337,2.232,3 1455 0027
     2.311,2.792,2.584,2.461,3.668,3.035,2.789,2.652,3.846,3.23,2.968,2. 1456 0028
     3821,3.84,3.37,3.113,2.963,3.767,3.475,3.25,3.104,1.543,1.515,1.491 1457 0029
     4,1.468,1.447,1.428,1.414,2.003,1.96,1.922,1.89,1.862,1.835,1.812,2 1458 0030
     5.51,2.426,2.357,2.301,2.253,2.21,2.173,3.151,2.967,2.836,2.734,2.6 1459 0031
     652,2.585,2.529,4.07,3.651,3.383,3.198,3.063,2.956,2.871,5.417,4.48 1460 0032
     76,3.997,3.69,3.477,3.318,3.196,5.793,5.081,4.509,4.119,3.844,3.64, 1461 0033
     83.484,4.855,4.893,4.637,4.339,4.084,3.879,3.714,4.14,4.37,4.391,4. 1462 0034
     9286,4.134,3.98,3.841,3.735,3.956,4.069,4.083,4.035,3.954,3.863/    1463 0035
      DATAAD/3.505,3.692,3.819,3.881,3.891,3.868,3.824,3.378,3.516,3.654 1464 0036
     1,3.712,3.77,3.769,3.767,5.417,4.486,3.997,3.69,5.82,4.795,4.229,3. 1465 0037
     2862,5.887,5.022,4.43,4.033,5.639,5.107,4.573,4.163,5.23,5.054,4.64 1466 0038
     33,4.251,4.855,4.893,4.637,4.339,2.47,2.73,3.077,3.551,4.192,4.958, 1467 0039
     45.659,5.946,5.793,2.482,2.678,2.916,3.205,3.549,3.916,4.304,4.637, 1468 0040
     54.855,2.503,2.656,2.833,3.033,3.255,3.496,3.725,3.949,4.14,2.535,2 1469 0041
     6.659,2.795,2.942,3.1,3.264,3.43,3.59,3.735,2.582,2.683,2.792,2.907 1470 0042
     7,3.026,3.146,3.27,3.391,3.505,2.641,2.731,2.822,2.913,3.003,3.097, 1471 0043
     83.191,3.284,3.378,25.22,30.03,35.61,40.25,44.37,48.45,24.47,30.12, 1472 0044
     936.7,42.16,47.08,51.94,22.95,29.45,36.95,43.16,48.74,54.28,20.18/  1473 0045
      DATAAE/27.65,36.2,43.19,49.41,55.58,16.12,24.74,34.5,42.3,49.15,55 1474 0046
     1.9,10.33,20.63,31.93,40.59,48.07,55.4,1.494,15.13,28.59,38.11,46.1 1475 0047
     25,53.98,-19.29,10.62,24.91,35.,43.5,51.78,0.,8.28,17.76,30.56,39.5 1476 0048
     33,48.33,0.,-55.,4.28,17.23,31.9,42.56,0.,0.,-9.11,3.79,17.,32.36,1 1477 0049
     41.93,-5.256,1.59,0.,.11,16.88,15.48,11.,4.18,-13.56,0.,0.,23.81,18 1478 0050
     5.76,13.42,-1.68,-17.82,0.,31.05,25.95,27.7,16.41,3.2,-10.31,38.05, 1479 0051
     632.9,33.42,32.03,20.87,9.05,44.95,39.75,39.57,39.66,34.95,24.46,10 1480 0052
     727.,1148.,1238.,1311.,1362.,1407.,1445.,962.3,1099.,1202.,1285.,13 1481 0053
     844.,1394.,1438.,869.,1026.,1142.,1235.,1301.,1357.,1405.,727.5,906 1482 0054
     9.9,1040.,1145.,1219.,1282.,1335.,533.3,742.1,895.5,1016.,1099./    1483 0055
      DATAAF/1170.,1230.,333.8,512.8,696.3,838.4,934.9,1015.,1083.,91.2, 1484 0056
     1290.2,434.2,603.4,720.3,814.5,893.8,0.,0.,153.4,302.5,433.2,558.1, 1485 0057
```

```
*********     SPHTDA     *********

     2653.3,343.4,72.4,-162.,20.,151.5,276.4,378.8,929.7,692.1,477.8,288   1486 0058
     3.3,134.,90.2,-120.,1489.,1269.,1082.,907.6,759.5,607.4,444.7,2037.   1487 0059
     4,1817.,1634.,1469.,1331.,1195.,1053.,2584.,2363.,2180.,2016.,1881.   1488 0060
     5,1751.,1620.,3135.,2911.,2726.,2563.,2428.,2299.,2173.,1.566,1.557   1489 0061
     6,1.78,1.77,2.059,2.037,2.377,2.333,2.926,2.724,1.091,3.21,2.46,5.2   1490 0062
     786,2.456,3.245,2.428,3.431,2.434,3.025,2.461,2.813,2.463,2.7,2.465   1491 0063
     8,2.638,2.47,2.612,35.34,16.83,13.18,11.55,10.58,9.919,9.429,54.51,   1492 0064
     926.45,20.19,17.34,15.62,14.45,13.58,64.37,38.42,29.61,25.31,22.66/   1493 0065
      DATAAG/20.81,19.43,56.31,49.66,39.93,34.49,30.94,28.41,26.46,9.931   1494 0066
     1,7.923,7.103,15.62,10.89,9.417,22.66,15.12,12.73,30.94,20.32,16.85   1495 0067
     2,7.93,6.276,5.692,5.433,5.279,5.173,10.9,7.934,6.883,6.417,6.14,5.   1496 0068
     395,15.15,10.31,8.585,7.82,7.364,7.052,17.43,13.27,10.71,9.57,8.89,   1497 0069
     48.424,5.173,4.912,4.796,4.726,4.679,7.052,6.278,5.935,5.731,5.591,   1498 0070
     59.796,8.258,7.575,7.167,6.891,0.,3.802,3.799,3.797,3.797,3.796,3.8   1499 0071
     686,3.875,3.87,3.867,3.865,4.01,3.979,3.965,3.956,3.951,4.216,4.14,   1500 0072
     74.106,4.085,4.072,4.576,4.408,4.333,4.289,4.259,5.191,4.855,4.707,   1501 0073
     84.618,4.558,6.202,5.582,5.308,5.144,5.032,7.785,6.713,6.238,5.954,   1502 0074
     95.761,10.15,8.396,7.619,7.155,6.839,13.53,10.8,9.588,8.865,8.372/    1503 0075
      DATAAH/18.14,14.09,12.29,11.21,10.47,24.18,18.44,15.86,14.32,13.26   1504 0076
     1,31.64,23.93,20.41,18.28,16.82,40.2,30.53,25.96,23.16,21.23,48.9,3   1505 0077
     27.93,32.38,28.89,26.45,55.98,45.39,39.27,35.23,32.33,59.18,51.55,4   1506 0078
     35.76,41.58,38.42,57.08,54.85,50.68,47.04,44.04/                      1507 0079
      DATAAI/2.819,2.811,2.809,2.809,2.808,2.808,2.808,3.032,2.963,2.95,   1508 0080
     12.944,2.94,2.938,2.936,3.624,3.255,3.186,3.155,3.136,3.124,3.114,5   1509 0081
     2.313,3.972,3.717,3.604,3.537,3.491,3.457,9.333,5.61,4.898,4.582,4.   1510 0082
     3394,4.265,4.17,17.15,8.823,7.203,6.481,6.049,5.754,5.536,29.47,14.   1511 0083
     422,11.11,9.705,8.862,8.284,7.855,2.808,2.807,2.807,2.938,2.931,2.9   1512 0084
     528,3.124,3.086,3.07,3.492,3.353,3.296,4.268,3.878,3.719,5.762,4.86   1513 0085
     65,4.498,8.3,6.533,5.809,2.807,2.807,2.806,2.806,2.806,2.806,2.931,   1514 0086
     72.925,2.923,2.922,2.921,2.92,3.085,3.055,3.044,3.039,3.036,3.034,3   1515 0087
     8.353,3.239,3.199,3.181,3.17,3.189,3.879,3.559,3.446,3.395,3.365,3.   1516 0088
     9345,4.867,4.129,3.867,3.751,3.682,3.72,6.541,5.077,4.557,4.327/      1517 0089
      DATAAJ/4.189,4.095,2.806,2.806,2.805,2.804,2.803,3.034,3.029,3.026   1518 0090
     1,3.024,3.022,3.345,3.294,3.271,3.257,3.247,4.095,3.861,3.757,3.694   1519 0091
     2,3.652,2.485,2.509,2.519,2.52,2.639,2.637,2.807,2.803,2.884,2.996,   1520 0092
     33.078,3.134,2.728,2.802,2.86,2.9,2.578,2.626,2.663,2.689,2.506,2.5   1521 0093
     438,2.562,2.578,2.482,2.504,2.52,2.53,2.48,2.489,2.5,2.509,1.602,1.   1522 0094
     5634,1.665,1.682,1.694,1.706,1.758,1.8,1.838,1.876,1.893,1.911,1.92   1523 0095
     69,1.947,1.961,2.005,2.046,2.084,2.215,2.216,2.217,2.219,2.229,2.23   1524 0096
     78,2.271,2.304,2.338,2.533,2.543,2.552,2.56,2.571,2.58,2.617,2.653,   1525 0097
     82.684,2.759,2.776,2.791,2.806,2.82,2.833,2.882,2.92,2.951,2.892,2.   1526 0098
     991,2.926,2.943,2.958,2.972,3.023,3.062,3.093,2.94,2.956,2.973/       1527 0099
      DATAAK/2.989,3.003,3.015,3.063,3.101,3.134,1.113,1.083,1.06,1.05,1   1528 0100
     1.353,1.335,1.325,1.312,1.496,1.498,1.494,1.491,1.572,1.591,1.601,1   1529 0101
     2.603,1.595,1.63,1.655,1.672,1.617,1.662,1.696,1.724,1.651,1.699,1.   1530 0102
     3738,1.771,1.693,1.741,1.782,1.819,1.757,1.807,1.85,1.888,1.173,1.1   1531 0103
     471,1.168,1.165,1.16,1.156,1.149,1.345,1.341,1.34,1.338,1.334,1.332   1532 0104
     5,1.329,1.456,1.455,1.454,1.453,1.452,1.45,1.449,1.517,1.519,1.521,   1533 0105
     61.523,1.525,1.526,1.527,1.551,1.552,1.556,1.561,1.564,1.569,1.572,   1534 0106
     71.583,1.571,1.571,1.575,1.581,1.587,1.591,1.61,1.591,1.584,1.584,1   1535 0107
     8.589,1.594,1.6,1.61,1.61,1.608,1.608,1.61,1.615,1.621,1.616,1.623,   1536 0108
     91.627,1.632,1.635,1.641,1.646,1.632,1.64,1.646,1.653,1.659,1.665/    1537 0109
      DATAAL/1.671,1.665,1.673,1.68,1.687,1.693,1.7,1.707,1.715,1.723,1.   1538 0110
     1731,1.738,1.744,1.751,1.757,1.583,1.571,1.571,1.575,1.597,1.578,1.   1539 0111
     2574,1.579,1.608,1.587,1.581,1.582,1.61,1.596,1.589,1.589,1.61,1.60   1540 0112
     34,1.598,1.598,1.61,1.61,1.608,1.608,1.485,1.511,1.534,1.557,1.577,   1541 0113
     41.604,1.621,1.621,1.61,1.495,1.516,1.534,1.552,1.569,1.583,1.596,1   1542 0114
     5,606,1.61,1.516,1.532,1.548,1.563,1.577,1.591,1.602,1.61,1.616,1.5   1543 0115
```

********* SPHTDA *********

```
     649,1.562,1.574,1.585,1.597,1.608,1.619,1.627,1.632,1.595,1.606,1.6  1544 01
     715,1.625,1.634,1.642,1.652,1.66,1.665,1.656,1.664,1.671,1.679,1.68  1545 01
     86,1.693,1.701,1.708,1.715,1.131,1.132,1.133,1.133,1.133,1.131,1.24  1546 01
     9,1.239,1.238,1.237,1.236,1.235,1.342,1.338,1.335,1.332,1.33,1.328/  1547 01
      DATAAM/1.417,1.412,1.409,1.406,1.404,1.402,1.464,1.462,1.46,1.459,  1548 01
     11.457,1.456,1.501,1.497,1.495,1.495,1.495,1.495,1.565,1.535,1.523,  1549 01
     21.52,1.52,1.522,1.754,1.78,1.564,1.546,1.541,1.541,1.943,2.3,1.71,  1550 01
     31.585,1.564,1.557,0.,2.4,1.808,1.653,1.594,1.573,0.,0.,1.62,1.683,  1551 01
     41.624,1.591,1.583,2.3,3.1,0.,1.632,1.608,1.33,1.976,2.608,1.56,0.,  1552 01
     50.,1.522,1.66,1.736,1.617,1.495,0.,1.527,1.589,1.679,1.663,1.609,1  1553 01
     6.56,1.522,1.563,1.608,1.648,1.627,1.601,1.52,1.551,1.579,1.613,1.6  1554 01
     722,1.609,1.151,1.152,1.152,1.152,1.153,1.153,1.153,1.243,1.242,1.2  1555 01
     841,1.241,1.24,1.24,1.24,1.327,1.326,1.325,1.324,1.323,1.322,1.321,  1556 01
     91.393,1.392,1.391,1.39,1.389,1.388,1.387,1.44,1.439,1.438,1.438/    1557 01
      DATAAN/1.437,1.436,1.436,1.478,1.474,1.473,1.472,1.471,1.471,1.47,  1558 01
     11.508,1.503,1.506,1.504,1.501,1.5,1.498,0.,0.,1.547,1.546,1.548,1.  1559 01
     2539,1.533,1.646,1.78,1.938,1.602,1.615,1.628,1.641,1.544,1.607,1.6  1560 01
     39,1.805,1.978,2.346,2.652,1.517,1.544,1.578,1.623,1.687,1.787,1.96  1561 01
     42,1.504,1.518,1.534,1.551,1.57,1.592,1.618,1.498,1.508,1.517,1.526  1562 01
     5,1.536,1.546,1.556,1.499,1.506,1.512,1.519,1.526,1.532,1.539,1.129  1563 01
     6,1.13,1.22,1.22,1.31,1.307,1.38,1.378,1.432,1.43,1.477,1.469,1.477  1564 01
     7,1.7,1.475,1.671,1.476,1.619,1.477,1.544,1.478,1.517,1.478,1.504,1  1565 01
     8.481,1.498,1.485,1.499,29.47,14.22,11.11,9.705,8.862,8.284,7.855,4  1566 01
     93.57,21.93,16.86,14.51,13.08,12.09,11.36,50.02,30.81,24.1,20.75/    1567 01
      DATAAO/18.65,17.17,16.06,44.26,38.61,31.58,27.55,24.89,22.97,21.48  1568 01
     1,8.3,6.533,5.809,13.08,9.068,7.803,18.65,12.54,10.56,24.89,16.66,1  1569 01
     23.88,6.611,5.077,4.597,4.327,4.189,4.095,10.09,6.517,5.6,5.191,4.9  1570 01
     347,4.78,12.6,8.52,7.052,6.395,6.001,5.732,16.8,10.95,8.824,7.865,7  1571 01
     4.289,6.894,4.095,3.861,3.757,3.694,3.652,5.732,5.06,4.761,4.584,4.  1572 01
     546,8.056,6.756,6.173,5.816,5.449,0.,2.817,2.813,2.812,2.811,2.811,  1573 01
     62.899,2.888,2.883,2.881,2.879,3.018,2.989,2.976,2.968,2.963,3.214,  1574 01
     73.142,3.111,3.092,3.079,3.55,3.394,3.325,3.284,3.255,4.119,3.809,3  1575 01
     8.672,3.59,3.534,5.044,4.476,4.225,4.074,3.972,6.478,5.504,5.071,4.  1576 01
     9813,4.637,8.587,7.014,6.314,5.895,5.609,11.54,9.137,8.061,7.417/    1577 01
      DATAAP/6.976,15.49,11.99,10.42,9.471,8.822,20.48,15.66,13.46,12.13  1578 01
     1,11.22,26.41,20.15,17.23,15.45,14.22,32.93,25.35,21.67,19.4,17.81,  1579 01
     239.24,30.92,26.61,23.87,21.93,44.14,36.29,31.69,28.62,26.38,46.18,  1580 01
     340.53,36.27,33.17,30.8,44.47,42.64,39.57,36.91,34.71/               1581 01
      END                                                                 1582 01
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - SPUMP

```
SPUMP* PROC
C
      REAL  NSG,NSS,NUZ,NU
C
      COMMON /SPUMP/ DI,EFFQ, H, DNS, XNS, NSG, NSS, NUZ, NU(2), U
C
 END
```

********* SUBRØUTINE STØCØN

```
C                    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                    * ROUTINE NAME - ROUTINE TO PACK THE FIRST          *
C                    *                WORD OF THE CONFIGURATION          *
C                    *                TABLE.                             *
C                    * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2       *
C                    * PROGRAMMER   - R. BOLLINGER 1943 102 26933        *
C                    * DATE CODED   - 3/11/70                            *
C                    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE STOCON(IDX)
C
C         ***** EXPLANATION OF THE CALLING SEQUENCE
C               *
C         ***** IDX - INDEX OF THE CONFIGURATION TABLE.
C
      INCLUDE CCNFIG
C
      DO 10 II = 1,6
      IF(ICNFIG(II).EQ.0) ICNFIG(II) = 0
      CONFIG(IDX,1) = GPBYTE(ICNFIG(II),6,CONFIG(IDX,1),II)
   10 CONTINUE
C
      RETURN
      END
```

******** SUBRØUTINE STØDTA

```
C
      SUBROUTINE STODTA
C
      INCLUDE CAPU
      INCLUDE CCNFIG
      INCLUDE CCNTRL
      INCLUDE CECLSS
      INCLUDE CFLRAT
      INCLUDE CHEX
      INCLUDE CHSORC
      INCLUDE CIOUNT
      INCLUDE CMATRL
      INCLUDE CNAMES
      INCLUDE CONST
      INCLUDE CPAGE
      INCLUDE CPUMP
      INCLUDE CTANK
      INCLUDE CTURBN
      INCLUDE TABLOK
C
C                 *****  PDP CAPU     *****
C
      DATA ((LAPU1(I,J),I=1,4),J=1,8)  /     'PROPELLANT TEMP. (DEG-R)',
     1   'OXYGEN LOADED       (LBS)'   ,     'HYDROGEN LOADED    (LBS)',
     2   'TOTAL LOADED        (LBS)'   ,     'PROP. USED BY APU  (LBS)',
     3   'OXY. MAX. FLOW  (LB/SEC)'    ,     'HYD. MAX. FLOW  (LB/SEC)',
     4   'WDOT OX-TURB-GG (LB/SEC)'    /
C
      DATA ((LAPU2(I,J),I=1,4),J=1,11) /     'WT. OF CIRCL. COMPRESSOR',
     1   'GAS GEN. CONSUMPTION    '    ,     'VOLUME OF STORAGE TANK  ',
     2   'AREA OF STORAGE TANK    '    ,     'WEIGHT OF STORAGE TANK  ',
     3   'H2 VENT - ABSORB TK LEAK'    ,     'WT. OF RESIDUAL IN TANK ',
     4   'WT. OF ACCUMULATOR TANK '    ,     'CAPACITY OF ACCUMULATOR ',
     5   'WT. OF RESID. IN ACCUM. '    ,     'TOT. WT. OF PROPL. REQD.'/
C
      DATA ((LAPU3(I,J),I=1,4),J=1,11) /     'TEMP. OF G.G. EXHAUST   ',
     1   'WTG. OF FLUID TO G.G.   '    ,     'SURF. AREA OF STO. TANK ',
     2   'VOLUME OF STORAGE TANK  '    ,     'HEAT LEAK INTO STO. TANK',
     3   'WT H2-ABSRB TK+PMP HT LK'    ,     'TOT.H2-VENTED FOR HT.LK.',
     4   'TOTAL WT. OF PROPELLANT '    ,     'STORAGE TANK RESIDUAL   ',
     5   'ACCUMULATOR CAPACITY    '    ,     'ACCUMULATOR TANK RESIDUL'/
C
C                *****  PDP CCNFIG  *****
C
      DATA ((LCNF1(I,J),I=1,4),J=1,6) / 'CONSUMER WEIGHT - LBS    ' ,
     1     'OXIDYZER SYSTEM WT. -LBS' , 'OXID INSULATION WT - LBS' ,
     2     'FUEL SYSTEM WT. - LBS   ' , 'FUEL INSULATION WT - LBS' ,
     3     'TOTAL SYSTEM WT. - LBS  ' /
C
C                 *****  PDP CCNTRL   *****
C
C                                          SUB CRIT.    SUPER CRIT.
      DATA ((INBLK(1,I,J),I=1,5),J=1,2) / 1,1,1,1,0,    1,1,0,1,0 /
      DATA ((INBLK(2,I,J),I=1,5),J=1,2) / 1,1,1,1,1,    1,1,0,1,0 /
      DATA ((INBLK(3,I,J),I=1,5),J=1,2) / 0,0,0,0,0,    0,1,0,0,0 /
      DATA ((INBLK(4,I,J),I=1,5),J=1,2) / 0,1,0,1,0,    0,1,1,0,1 /
      DATA ((INBLK(5,I,J),I=1,5),J=1,2) / 0,0,0,0,0,    0,0,1,0,0 /
```

```
      DATA (KSUBC(1,I),I=1,NBRSR) / 6, 4, 10, 9, 8, 1, 10, 11, 2/
      DATA (KSUBC(2,I),I=1,NBRSR) / 6, 3, 4, 10,11, 2,  0,  0, 0/
      DATA (KSUBC(3,I),I=1,NBRSR) / 7, 0, 0,  0, 0, 0,  0,  0, 0/
      DATA (KSUBC(4,I),I=1,NBRSR) / 5, 4, 0,  0, 0, 0,  0,  0, 0/
      DATA (KSUBC(5,I),I=1,NBRSR) / 6, 4, 10, 9, 8, 1, 10, 11, 2/
      DATA JAPUS(1,1),JAPUS(1,2) / 4, 3/
      DATA JAPUS(2,1),JAPUS(2,2) / 3, 4/
      DATA NAMSYS / 'ACP','APU','EC/','FUE','OMS' /
C
C                ***** PDP CFLRAT *****
C
      DATA ((LFRT(I,J),I=1,3),J=1,7)/
     1  'WDOT OX-TURB.-G.G.','WDOT HY-TURB.-G.G.','WDOT BOTH TURB.-GG',
     2  'WDOT OXY HEX.-G.G.','WDOT HYD HEX.-G.G.','WDOT BOTH HEX.-G.G',
     3  'TOTAL FLOWRATE  **'/
C
C                ***** PDP CHEX     *****
C
      DATA  ((UNAM(I,J),I=1,2),J=1,4) /
     1  '   BOILING  ','SUP-CRITICAL','PARALLEL-FLO','COUNTER-FLOW'/
      DATA  ((LHX1(I,J),I=1,4),J=1,10)/
     1  'THERML CONDUCTANCE RATIO','HOT FLUID FLOW RATE     ',
     2  'COLD FLUID DELTA - P    ','CAPACITY RATIO          ',
     3  'NUMBER OF TRANSFER UNITS','COMPUTED VALUE OF UA    ',
     4  'COMPUTED VALUE OF W/UA  ','WEIGHT OF SUBUNIT       ',
     5  'WEIGHT OF HEAT EXCHANGER','HEX SUBUNIT TYPE  ***   '/
      DATA  ((LHX2(I,J),I=1,4),J=1,14)/
     1  'COLD FLUID INLET TEMP   ','COLD FLUID OUTLET TEMP  ',
     2  'COLD FLUID SPECIFIC HEAT','COLD FLUID FLOW RATE    ',
     3  'HOT FLUID INLET TEMP    ','HOT FLUID OUTLET TEMP   ',
     4  'HOT FLUID SPECIFIC HEAT ','HOT FLUID FLOW RATE     ',
     5  'COLD SIDE EFFECTIVENESS ','HOT SIDE EFFECTIVENESS  ',
     6  'TOTAL EFFECTIVENESS     ','HEAT EXCHANGER UA/A-WALL',
     7  'HEAT EXCHANGER DIAMETER ','HEAT EXCHANGER LENGTH   '/
      DATA LHX3 / 'HEAT EXCHANGER CHARACTERISTICS' /
C
C                ***** PDP CHSORC    *****
C
      DATA ((LHS1(I,J),I=1,5),J=1,6) /
     1'GAS GENERATOR CHARACTERISTICS ','GAS GEN. FLOW RATE - (LB/SEC) ',
     2'GAS GEN. PROPELLANT WGT.-(LBS)','GAS GENERATOR WEIGHT - (LBS)  ',
     3'CUMULATIVE GAS GEN. PROP. WTG.','WEIGHT OF HEX-GAS GEN. ASSY.  '/
C
      DATA ((LHS2(I,J),I=1,5),J=1,14) /
     1'SPEC.HEAT AVAILABLE-(BTU/LB-R)','TOTAL HEAT REQUIRED - (BTU)   ',
     2'HOT FLUID REQUIRED - (LBS)    ','CUMULATIVE HEAT REQD. - (BTU) ',
     3'CUMULATIVE HOT FLUID - (LBS)  ','WASTE HEAT UTILIZATION DATA   ',
     4'MAX HOT FLUID FLO-RATE(LBS/HR)','CYCLE MAX REQD ENERGY - O2 HEX',
     5'CYCLE MAX REQD ENERGY - N2 HEX','CYCLE MAX REQD ENERGY- O2 TANK',
     6'CYCLE MAX REQD ENERGY- N2 TANK','TOTAL MAX ENERGY - HEX + TANKS',
     7'TOTAL ENERGY FOR MISSION SPAN ','TOTAL ENERGY REQMT - KW/HRS   '/
C
C                ***** PDP CIOUNT    *****
C
      DATA IOUNIT/14,21,22,23,19,29,15,16,17,18,25,26,27,28/
      DATA  IIN,IOT / 5,6 /
C
C                ***** PDP CMATRL    *****
```

```
C
      DATA(RHOL(I),I=1,5)/501.120,176.256,169.944,511.488,276.480/
      DATA (RHOI(I),I=1,9)/2.14,2.45,2.80,0.836,0.59,0.65,5.0,2.2,1.0/
      DATA (RHOIS(I),I=1,9)/0.0428,0.0490,0.0280,0.027867,0.01475,
     1 0.021667,5.0,2.20,1.0/
      DATA(MINTHK(I),I=1,15)/0.020,0.025,0.028,0.020,0.016,0.035,0.058,
     1                     0.065,0.042,0.020,0.049,0.083,0.095,0.049,0.035/
C
C                    *****   PDP CNAMES   *****
C
      DATA(FNAME(I),I=1,18)/'GAS   ','ENGINE','LINE  ','CONTRL','FITING'
     1                     ,'TAP   ','TEE   ','ELBOW ','VALVE ','REG   '
     2                     ,'ACCUM ','TANK  ','PUMP  ','HEX   ','TRBINE'
     3                     ,'F-CELL','EC/LSS','END   '/
C
      DATA ((LO(I,J),I=1,9),J=1,15)/
     1         '  ***** S Y S T E M   C O N F I G U R A T I O N ***** ',
     2         '              ***** E N G I N E   D A T A *****        ',
     3         '        ***** A C C U M U L A T O R   D A T A *****    ',
     4         '     ***** H E A T   E X C H A N G E R   D A T A ***** ',
     5         '   ***** H I G H   P R E S   P U M P   D A T A *****   ',
     6         '         ***** D U T Y   C Y C L E   D A T A *****     ',
     7         '         ***** H E A T   S O U R C E   D A T A *****   ',
     8         '              ***** T U R B I N E   D A T A *****      ',
     9         '                ***** M O T O R   D A T A *****        ',
     T         '*** T A N K   C O N F I G U R A T I O N   D A T A ****',
     1         '     ***** L O W   P R E S   P U M P   D A T A *****   ',
     2         '                ***** T A N K   D A T A *****          ',
     3         ' ***** A U X I L I A R Y   P O W E R   U N I T *****   ',
     4         '         ***** F U E L   C E L L   D A T A *****       ',
     5         '             ***** E C / L S S   D A T A *****         '/
C
      DATA ((L1(I,J),I=1,21),J=1,2)/'     COMP    COMP    FUNC.  NUMB.  NU
     1MB.  MATRL.  FLOW FRICTION   LINE LENGTH     LINE     INSULATION
     2 INSULATION   NO. LAYERS ','     NAME   CODE    TYPE    OPER.  STBY.
     3  TYPE      COEFICIENT      OR L-OVER-D   DIAMETER       TYPE       T
     4HICKNESS   INSULATION '/
C
      DATA ((L2(I,J),I=1,3),J=1,7)/'NUMBER OF ENGINES ',
     1       'GAS INLET TEMP.    ','GAS INLET PRES.    ',
     2       'ENGINE THRUST      ','CHAMBER PRES.      ',
     3       'EXPANSION RATIO    ','MIXTURE RATIO      '/
C
      DATA ((L3(I,J),I=1,4),J=1,3)     /    'OPERATING TEMP. (DEG R) ',
     1   'TANK VOLUME (CU. FT.)   '   ,     'NOMINAL OPER. DELTA PRES'/
C
      DATA((L4(I,J),I=1,4),J=1,11)/ 'HEX HOT INLET TEMP.         ',
     1   'HEX HOT OUTLET TEMP.      ','HEX COLD INLET TEMP.       ',
     2   'HEX COLD OUTLET TEMP.     ','HEX HOT INLET PRES.        ',
     3   'HEX HOT OUTLET PRES.      ','HEX COLD INLET PRES.       ',
     4   'HEX COLD OUTLET PRES.     ','HEX HOT SIDE DELTA-P       ',
     5   'HEX COLD SIDE DELTA-P     ','HEX GAS GEN. O/F RATIO     '/
C
      DATA ((L5(I,J),I=1,4),J=1,5)/ 'TYPE                        ',
     1   'EFFICIENCY                ','NET + SUCTION HEAD          ',
     2   'SHAFT SPEED               ','ESTIMATED DELTA PRES.       '/
C
      DATA ((L6(I,J),I=1,4),J=1,4)/ 'PUMP EFFICIENCY             ',
```

```
********     STODTA     ********

        1    'NET POS. SUCTION HEAD    ','PUMP PRESSURE RISE       ',
        2    'PUMP FLOW RATE           '/
C
        DATA ((L7(I,J),I=1,4),J=1,23)     /
        1    '                         ',      'MAXIMUM DIAMETER (FT)    ',
        2    'INITIAL FLUID LOAD (OPT)',       'PERCENT ULLAGE VOLUME    ',
        3    'HEAT FLUX (BTU/HR-FT**2)',       'INITIAL TEMPERATURE (R)  ',
        4    'INITIAL PRESSURE        ',       'VENTING PRESSURE         ',
        5    'MATERIAL TYPE           ',       'INSULATION TYPE          ',
        6    'INSULATION THICKNESS    ',       'ACQUISITION TYPE         ',
        7    'PRESSURIZATION TYPE     ',       'OPERATING PRESS. (PSIA)  ',
        8    'PRESSURANT GAS TEMP. (R)',       'HEX DELTA PRESS. (PSIA)  ',
        9    'PUMP DELTA PRESS. (PSIA)',       'HEX OUTLET TEMP. (R)     ',
        T    'P SUB C OF GAS GEN (PSIA',       'GAS GEN OUTLET TEMP (R)  ',
        1    'GAS GEN MIXTURE RATIO   ',       'NUMBER OPERATING (NOP)   ',
        2    'NUMBER INSULATION LAYERS'     /
C
        DATA ((L8(I,J),I=1,4),J=1,3)/ '( ENG. DEG. DUE TO MIB   ',
        1    ' ACTIVE          INACTIVE ',' TIME              TIME   '/
C
        DATA ((L9(I,J),I=1,4),J=1,5)/ 'TURBINE EFFICIENCY        ',
        1    'TURBINE INLET TEMP.      ','TURBINE OUTLET TEMP.      ',
        2    'TURBINE MIXTURE RATIO    ','TURBINE GAS GEN. PSUBC    '/
C
        DATA((L10(I,J),I=1,4),J=1,5)/ 'HEAT SOURCE TYPE          ',
        1    'HEAT SOURCE MIX. RATIO   ','HEAT SOURCE OUTLET TEMP.',
        2    'HEAT SOURCE AVAIL.ENERGY','HEAT SOURCE PRESSURE      '/
C
        DATA((L11(I,J),I=1,4),J=1,20)/'NUMBER OF APU UNITS       ',
        1    'HORSEPOWER PER UNIT      ','TURBINE MIXTURE RATIO     ',
        2    'APU GAS GEN. INLET PRESS','TURBINE INLET TEMP.       ',
        3    'HEX EXHAUST DISCHGE.TEMP','H2-SUP.GAS GEN.MIX-RATIO',
        4    'O2-SUP.GAS GEN.MIX-RATIO','H2-SUP.GAS GEN.EXIT TEMP',
        5    'O2-SUP.GAS GEN.EXIT TEMP','H2-TANK RESID.VAP.TEMP.  ',
        6    'O2-TANK RESID.VAP.TEMP.  ','SYSTEM ENVIRONMENT TEMP.',
        7    'FUEL MIX-RAT.SUP.GAS-GEN','FINAL H2 TANK PRESSURE    ',
        8    'FINAL O2 TANK PRESSURE   ','FINAL H2 TANK TEMP.       ',
        9    'FINAL O2 TANK TEMP.      ','TEMP.EX.PROD.SUP.GAS-GEN',
        T    'DELTA-P TANK CIRC. PUMP '/
C
        DATA((L12(I,J),I=1,4),J=1,28)/'NUMBER OF FUEL CELLS OP.',
        1    'FUEL CELL MIXTURE RATIO ','SP.REACTANT CONSUMPTION ',
        2    'F.C.HEAT REJECTION RATE ','SP.WGT. OF FUEL CELL     ',
        3    'FUEL CELL NOM. TEMP.- O2','FUEL CELL NOM. TEMP.- H2',
        4    'HOT FLUID INLET TEMP.    ','HOT FLUID OUTLET TEMP.   ',
        5    'O2 TANK FILL DENSITY     ','H2 TANK FILL DENSITY     ',
        6    'EST. O2 TANK VENT QUANT.','EST. H2 TANK VENT QUANT.',
        7    'FUEL CELL OPER. PRESSURE','NOM.FUEL CELL OPER.POWER',
        8    'O2 TANK LO-PRES. SETTING','H2 TANK LO-PRES. SETTING',
        9    'O2 TANK VAC.JAC. ANNULUS','H2 TANK VAC.JAC. ANNULUS',
        T    'O2 TANK MAXIMUM DIAMETER','H2 TANK MAXIMUM DIAMETER',
        A    'FUEL CELL VOLTAGE (AVG) ','FUEL CELL O2 PURGE RATE ',
        B    'FUEL CELL H2 PURGE RATE ','FUEL CELL O2 PURGE TIME ',
        C    'FUEL CELL H2 PURGE TIME ','O2 PURGE INTERVAL-AMPHRS',
        D    'H2 PURGE INTERVAL-AMPHRS'/
C
        DATA((L13(I,J),I=1,4),J=1,28)/'MISSION DURATION - DAYS ',
        1    'NUMBER OF MEN IN CREW   ','AIRLOCK REPRESSURIZINGS ',
```

******** STODTA ********

```
     2   'DAYS SUPPLY RESERVE GAS ','02 CONSUMED PER MAN-DAY ',
     3   'VEHICLE GAS LEAKAGE RATE','DELIVERED 02 NOM. TEMP. ',
     4   'DELIVERED N2 NOM. TEMP. ','02 FILL DENSITY         ',
     5   'N2 FILL DENSITY         ','02 TANK FINAL TEMP      ',
     6   'N2 TANK FINAL TEMP.     ','02 TANK FINAL PRESSURE  ',
     7   'N2 TANK FINAL PRESSURE  ','LSS ENVIRONMENT TEMP.   ',
     8   'CABIN OR AIRLOCK VOLUME ','02-HEX INLET LINE DIAM. ',
     9   'N2-HEX INLET LINE DIAM. ','HEX HEATER ENERGY RATING',
     T   'TNK HEATER ENERGY RATING','DELIVERED 02 PRESSURE   ',
     A   'DELIVERED N2 PRESSURE   ','02 TANK HEATER DIAMETER ',
     B   'N2 TANK HEATER DIAMETER ','02 TANK HEATER LENGTH   ',
     C   'N2 TANK HEATER LENGTH   ','02 TANK LOW-PRESS. LIMIT',
     D   'N2 TANK LOW-PRESS. LIMIT'/
C
      DATA ((JFLUID(I,J),I=1,2),J=1,3)/'   OXYGEN   ','  HYDROGEN  ',
     1                                 '  NITROGEN  '/
      DATA ((KFLUID(I,J),I=1,2),J=1,2)/'  OXIDYZER  ','    FUEL    '/
C
C                 *****   PDP CONST    *****
C
      DATA GRAVTY,PI,PI203 / 32.172 ,3.14159265 ,2.0943951 /
C
C                 *****   PDP CPAGE    *****
C
      DATA MAXLIN,JNUM  ,OPTLUN/50,'AT4307',6 /
      DATA PTITLE/'     THE INTEGRATED MATH MODEL      '/
C
C                 *****   PDP CPUMP    *****
C
      DATA ((LPP1(I,J),I=1,3),J=1,6) /
     1'TEMPERATURE       ','PRESSURE          ','FLOW RATE         ',
     2'DELTA-PRESSURE    ','NPSH AVIALABLE    ','DENSITY OF FLUID  '/
      DATA LPP2 / 'NUMBER OF STAGES REQD.  ' /
      DATA ((LPP3(I,J),I=1,3),J=1,6) /
     1'COMPUTED NPSP REQD','COMPUTED PUMP EFF.','COMPUTED PUMP VOL.',
     2'COMPUTED PUMP WGT.','COMPUTED PUMP PWR.','COMPUTED PUMP SPD.'/
      DATA LPP4 / 'SELECTED PUMP OPTION    ' /
      DATA LPP5 / 'PUMP CHARACTERISTICS    '/
C
C                 *****   PDP CTANK    *****
C
      DATA ((LTZ1(I,J),I=1,3),J=1,14)/
     1   'NUMBER OF TANKS   ','MATERIAL TYPE     ','INSULATION TYPE   ',
     2   'FLUID WGT. (TOTAL)','FLUID VOLUME /TANK','DIAMETER (FT)/TANK',
     3   'SURFACE AREA /TANK','TANK VOLUME / TANK','TANK WGT. (LB) TOT',
     4   'INSUL. THICKNESS  ','INSUL. WT (LB) TOT','HEAT LEAK BTU/H/FT',
     5   'GAS RESIDUALS WT. ','WGT ADDED CYL SECT'/
      DATA ((LTZ2(I,J),I=1,2),J=1,3)
     1       / 'SURF TENSION' , 'POSITV DISPL' , 'DIELCTROPHOR' /
      DATA ((LTZ3(I,J),I=1,3),J=1,4)                / 'TYPE ACQ. DEVICE  ',
     1   'DEVICE WT. (LBS)  ','TRAPPED BY DEVICE ','RESID. PROPELLANT '/
C
C                 *****   PDP CTURBN   *****
C
      DATA LTBN1 /'TURBINE CHARACTERISTICS '/
      DATA ((LTBN2(I,J),I=1,5),J=1,6) /
     1'TURBINE ROTOR MEAN DIAMETER   ','WGT. OF PWR. TRANSMISSION ASSY',
     2'WGT. OF TURBINE ROTOR         ','WGT. OF MANIFOLD AND NOZZLE   ',
```

```
********     STODTA     ********

      3'WEIGHT OF INDUCER              ','WEIGHT OF TURBINE ASSY.        '/
C
C                    *****   PDP TABLOK    *****
C
       DATA  NTBID / 1, 2, 7, 8, 4, 3, 6, 5,21, 9,
      1              10,10,17,13,13,16,13,21,27,28,
      2              29,30,31,32,21,32,35,32,38,35,
      3              10,13,16,13,41,41,32,32,41,18,
      4              19,20,21,44,45,46, 0, 0, 0, 0/
       MIPE = 0.
       MIENTH = 0.0
C
       RETURN
C
       END
```

******** PRØCEDURE DEFINITIØN PRØCESSØR - TABLØK

```
TABLOK* PROC
C
      REAL  MIPE
C
      COMMON /TABLOK/ XTAB(7), NTBID(50)
C
 END
```

********* SUBRØUTINE TANK

```
C                   * * * * * * * * * * * * * ** * * * * * * * * * * *
C                   * ROUTINE NAME - TANK PRES. AND WT. DUTY          *
C                   *                CYCLE HISTORY                    *
C                   * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                   * PROGRAMMER   - R. VERNON, 6263 , 562, 24385*
C                   * DATE CODED   - 5/12/72                          *
C                   * * * * * * * * * * * * * ** * * * * * * * * * * *
C
      SUBROUTINE TANK
C
      LOGICAL  JP, DIAG, PAGE
C
      INCLUDE CACCUM
      INCLUDE CDCYCL
      INCLUDE CENG
      INCLUDE CHEX
      INCLUDE CIOUNT
      INCLUDE CMATRL
      INCLUDE CMOTOR
      INCLUDE CNAMES
      INCLUDE CTANK
      INCLUDE TABLOK
C
      DIMENSION PRES(ICDL1,2,1)
      DIMENSION PWTOT(2)
      DIMENSION THED(2)
      DIMENSION WDOTX(MHX,2)
C
      EQUIVALENCE (WDOTCF,WDOTX)
C
      DATA THED(1), THED(2)/170.0,40.0/
C
C         ***** INITIALIZE THE PROGRAM COUNTERS
C             *
C             * IG - GAS TYPE FLAG
C             *      IG = 1 FOR O2
C             *           2 FOR H2
C             *
C             *      IT = 1 (ONLY)   REVISED  6-30-72
C             *
C             * IP - COUNTER FOR THOSE VARIABLE STORED FOR EACH ELEMENT
C             *      OF THE DUTY CYCLE. TANK PRES., HE PRES
C             *
C             * IW - COUNTER FOR THOSE VARIABLE STORED FOR EACH COAST
C             *      OF THE DUTY CYCLE. VENTED GAS WEIGHT,
C             *
C             * IF - COUNTER FOR THOSE VARIABLE STORED FOR EACH BURN
C             *      OF THE DUTY CYCLE. HE FLOW RATE, PRESSUREANT GAS
C         *****      FLOW RATE, ENERGY FLOW.
C
      IF (DIAG(0,6HTANK  )) WRITE (IOT,6000) NDCYCL,NOP,SATYPE,SITYPE,
     1  SMTYPE,SPTYPE,SMDIAM,SVLFLD,SULGPC,SHFLUX,SITEMP,SIPRES,SVPRES,
     2  SITHIK,SOPRES,SPGTEM,SHDELP,SPDELP,SHOTEM,SGGPC,SGOTEM,SGMRAT
C
      IG = 0
      IT = 1
   30 IG = IG + 1
```

********* TANK *********

```
      IBURN = 0
      ICOAST = 0
      IF = 0
      IP = 0
      ISW = 0
      IW = 0
      WLRT(IG,IT)=0.
      WHESUM = 0.0
      GO TO (80,80,70),IG
C
   70 JP = DIAG(1,6HTANK  )
      RETURN
C
C            ***** CALCULATE THE EFFECTIVE TANK DENSITY RHO.
C
80    T=TSAT(SIPRES(IG,IT),IG)
      PWTOT(IG) = WPTOT(IG)
      WP = WPTOT(IG)
      CALL RHOLIQ(T,IG,RHOLI)
      VLIQ=WP/RHOLI
      PVOL=SVOL(IG,IT)-VLIQ
C                   CALC. RHO OF GAS
      CALL GSDNST (IG,T,SIPRES(IG,1),RHOG)
      WPV=RHOG*PVOL
      WPT=WP+WPV
      RHO=WPT/SVOL(IG,IT)
      RHOP=RHO
      PHE=0.
      PPV=SIPRES(IG,IT)
      WHE=0.
      WTOT = WPT
C
      IF(DIAG(2,6HTANK-0))  WRITE (6,6005) T,WP   ,WP,RHOLI,VLIQ,
     1 SVOL(IG,IT),PVOL,RHOG,PPV,WTOT
C
C            ***** LOOK UP THE INITIAL ENERGY LEVEL
C
      KTAB = 0
      IF (IG .EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
      CALL FINTAB (NTBID(26)+IG+KTAB)
      XTAB(1) = RHO
      XTAB(2) = SIPRES(IG,IT)
      ENERGY = MIPE(2,XTAB)
C
      IF(DIAG(2,6HTANK1 ))WRITE (6,6033) IG,IT,IP,IW,IF,
     1   RHOP,ENERGY,WP,WPV,WPT,SIPRES(IG,IT)
C
      JP = PAGE(0)
C
      WRITE (IOT,7000)
      WRITE (IOT,7001)
      WRITE (IOT,7002) (JFLUID(I,IG),I=1,2), T, SIPRES(IG,IT), WP, WPV,
     1                 WPT, WHE, WTOT, VLIQ, PPV, PHE, PVOL, SVOL(IG,IT)
     2                ,RHOP, ENERGY
C
C            ***** DO ENERGY BALANCE FOR FIRST COAST FOR ALL FOUR SYSTEMS,
C            ***** AND ALL COASTS FOR SYSTEM 1 (SELF PRESSURIZATION).
C
```

********   TANK   *********

```
       90 IP = IP + 1
          IF(IP.LT.NDCYCL) GO TO 110
          IF(IP.EQ.NDCYCL) GO TO 115
      100 ISPTYP = SPTYPE(IG,IT)
          IP = IP - 1
          ICOAST = 0
          IBURN = 0
          GO TO 1010
C
      110 ENERGY = ENERGY + SHRATE(IG,IT)*DCYCLE(IP+1)/WPT
C
          ICOAST = ICOAST + 1
          IF(DIAG(2,6HTANK-A)) WRITE(6,6030) ICOAST,ISW
C
          IF(PAGE(35)) WRITE (IOT,7017)
C
          WRITE (IOT,6030) ICOAST, ISW
          WRITE (IOT,7003)
C
          GO TO 116
C
      115 ENERGY = ENERGY + SHRATE(IG,IT) * 300.0/WPT
          IF(DIAG(2,6HTANK-F)) WRITE (6,6031)
C
          IF(PAGE(17)) WRITE (IOT,7017)
C
          WRITE (IOT,6031)
          WRITE (IOT,7013)
C
      116 CONTINUE
C
C              ***** LOOK UP RESULTING PRES.
C
          CVP=CSUBV(T,PPV,IG)
      118 CVH=0.745
          RATIO=(WPT*CVP)/(WPT*CVP+WHE*CVH)
          KTAB = 0
          IF (IG .EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
          CALL FINTAB (NTBID(27),IG,KTAB)
          XTAB(1)=RHOP
          XTAB(2)=ENERGY*RATIO
          PPV=MIPE(2,XTAB)
          T=TSAT(PPV,IG)
C                         CALC. RHO OF GAS
          CALL GSDNST (IG,T,PPV,RHOG)
          IF(WHE.LE.0.) GO TO 119
          CALL ZFIND(T,PHE,17,ZHE)
          PHE=FINDR(17)*WHE*ZHE*T/(144.*PVOL)
      119 PRES(IP,IG,IT)=PPV+PHE
          WPV=RHOG*PVOL
          WR=WPT-WPV
C
          IF(DIAG(2,6HTANK2 )) WRITE (6,6060) IG,IT,IP,IW,IF,
         1    RHOP,ENERGY,CVP,CVH,RATIO,PPV,PHE,PRES(IP,IG,IT)
         2,WP, WPV, RHOG
C
          IF(DCYCLE(IP+1).LT.0.0) DCYCLE(IP+1) = 300.0
C
```

```
      WRITE (IOT,7004) (JFLUID(I,IG),I=1,2), T, DCYCLE(IP+1), WP, WPV,
     1 WHE, PPV, PHE, PRES(IP,IG,IT),ENERGY
C
C     ***** IF THE RESULTING PRES. IS GREATER THAN THE VENTING
C         * PRES., COMPUTE THE WEIGHT OF THE VENTED MATERIAL AND
C     ***** SET PRES TO THE VENTING PRES.
C
      IW = IW + 1
      SVWT(IW,IG,IT) = 0.0
      IF(PRES(IP,IG,IT) - SVPRES(IG,IT))130,130,120
  120 PV=SVPRES(IG,IT)
      PI=PRES(IP,IG,IT)
      V=SVOL(IG,IT)
      CALL VENT(0.,WHE,WPV,WP,T,PV,PI,V,IG,PPV,RHOP)
      SVWT(IW,IG,IT)=WPT-WP-WPV
      PWTOT(IG) = WP
      WPT=WP+WPV
      WTOT = WPT + WHE
      PHE=PV-PPV
      PRES(IP,IG,IT) = SVPRES(IG,IT)
C
C
C
C     ***** CALCULATE A NEW TANK DENSITY
C
      RHOP=WPT/SVOL(IG,IT)
      CVP=CSUBV(T,PPV,IG)
  128 CVH=0.745
      RATIO=(WPT*CVP)/(WPT*CVP+WHE*CVH)
C
C
C     ***** LOOK UP NEW ENERGY LEVEL
C
      KTAB = 0
      IF (IG .EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
      CALL FINTAB (NTBID(28)+IG+KTAB)
      XTAB(1)=RHOP
      XTAB(2)=PPV
      ENERGY=MIPE(2,XTAB)/RATIO
C
      IF(DIAG(2,6HTANK3 )) WRITE (6,6035) IG,IT,IP,IW,IF,
     1    SVWT(IW,IG,IT),WTOT,PPV,WP,PHE,PRES(IP,IG,IT),RHOP,RATIO,ENERGY
C
      WRITE (IOT,7005)
      WRITE (IOT,7006) SVPRES(IG,IT), SVWT(IW,IG,IT), WP, WPV, WHE,
     1 WTOT, PPV, PHE, PRES(IP,IG,IT), ENERGY
C
C     ***** UPDATE THE DUTY CYCLE COUNTER AND DO THE DIFFERENT
C     ***** SYSTEMS.
C
  130 IP = IP + 1
      IF(IP.GT.NDCYCL) GO TO 100
      ISW = SPTYPE(IG,IT)
C
      IBURN = IBURN + 1
      IF(DIAG(2,6HTANK-B)) WRITE(6,6020) IBURN,ISW
C
      WRITE (IOT,6020) IBURN, ISW
```

```
*********    TANK    *********

            WRITE (IOT,7007)
      C
            GO TO (1000,1000,3000),ISW
      C
      C         ***** DO ENERGY BALANCE FOR ALL BURNS
      C
       1000 CALL FINTAB (NTBID(29)+IG)
            H = MIPE(1,PPV)
            WDOTJM = WDOTJ(IBURN,IG) * PWTOT(IG)/WTTOT(IG)
            E=ENERGY*WTOT-H*WDOTJM*DCYCLE(IP-1)
      C
            IF(DIAG(2,6HTANK4A)) WRITE(6,6040) IG,IT,IP,IW,IF,
           1   H,WDOTJ(IBURN,IG),PWTOT(IG),WTTOT(IG),WDOTJM,ENERGY,WTOT,
           2   DCYCLE(IP-1),E
      C
            WRITE (IOT,7008) (JFLUID(I,IG),I=1,2), DCYCLE(IP-1),
           1                 WDOTJ(IBURN,IG), WTTOT(IG), PWTOT(IG), ENERGY, E,
           2                 WDOTJM
      C
            WOUT=WDOTJM*DCYCLE(IP-1)
            WTOT=WTOT-WOUT
            WPT=WPT-WOUT
            ENERGY=E/WTOT
      C
            WTTOT(IG) = WTTOT(IG) - WDOTJ(IBURN,IG) * DCYCLE(IP-1)
      C
            PWTOT(IG) = WPT
      C
      C         ***** CALCULATE NEW TANK DENSITY
      C
            RHOP=WPT/SVOL(IG,IT)
      C
            IF(DIAG(2,6HTANK4B)) WRITE(6,6050) IG,IT,IP,IW,IF,
           1  WOUT,WTOT,WPT,ENERGY,WTTOT(IG),PWTOT(IG),E,SVOL(IG,IT),RHOP
      C
      C         ***** LOOK UP THE RESULTING PRESSURE
      C
            KTAB = 0
            IF (IG .EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
            CALL FINTAB (NTBID(30)+IG+KTAB)
            XTAB(1)=RHOP
            RATIO=(WPT*CVP)/(WPT*CVP+WHE*CVH)
            XTAB(2)=ENERGY*RATIO
            PPV=MIPE(2,XTAB)
            IF(ISW.EQ.1) PRES(IP,IG,IT)=PPV
      C
            IF(DIAG(2,6HTANK4C)) WRITE(6,6061) IG,IT,IP,IW,IF,
           1   RHOP, ENERGY, RATIO, PPV, PRES(IP,IG,IT)
      C
            WRITE (IOT,7009)
            WRITE (IOT,7010) WOUT, WTOT, WPT, WTTOT(IG), RHOP, PPV, ENERGY
      C
            GO TO (2000,2000,3000),ISW
      C
      C         ***** DO THE FINAL CALCULATIONS FOR SYSTEM 1 AND PART SYSTEMS 2+3
      C
      C         ***** SUM UP THE VENTED GAS WEIGHT
      C
```

```
*********     TANK    *********

      1010 WVSUM = 0.0
           DO 1020 II = 1,IW
           WVSUM = WVSUM + SVWT(II,IG,IT)
      1020 CONTINUE
      C
           SWVTOT(IG,IT) = WVSUM
      C
      C        ***** CALCULATE THE PROPELLANT PARENT GAS RESIDUAL.
      C
           TFINAL = TSAT(PPV,IG)
      C                    CALC. RHO OF GAS
           CALL GSZDNS(IG,TFINAL,PPV,RHOIG,ZIG)
           WGR(IG,I) = RHOIG*PVOL + WHE
           WLRT(IG,I) = WP
      C
           IF(DIAG(2,6HTANK5 )) WRITE(6,6070) IG,IT,IP,IW,IF,WVSUM,TFINAL,Z,
          1 SWVTOT(IG,IT),WGR(IG,IT),WLRT(IG,IT)
      C
           WRITE (IOT,7014) TFINAL, SWVTOT(IG,IT), WGR(IG,IT), WLRT(IG,IT)
      C
           GO TO (30,2010,3010),ISPTYP
      C
      C        ***** CALCULATE PRESSURANT NEEDED FOR THIS BURN
      C
      2000 T=TSAT(PPV,IG)
      C                    CALC. RHO OF GAS
           CALL GSZDNS (IG,T,PPV,RHOG,Z)
           CALL RHOLIQ(T,IG,RHOLI)
           PVOL=SVOL(IG,IT)*(RHOLI-RHOP)/(RHOLI-RHOG)
           VLIQ=SVOL(IG,IT)-PVOL
           WP=VLIQ*RHOLI
           WPV=RHOG*PVOL
           IF(ISW.EQ.1) GO TO 90
           THE = THED(IG)
           IF=IF+1
           IF(PHE.LE.0.) GO TO 2004
           CALL ZFIND(T,PHE,I7,ZI)
           RHE=WHE/PVOL
           PHE=RHE*ZI*FINDR(I7)*T/144.
      2004 PTOT=PPV+PHE
           IF(PTOT.LT.SOPRES(IG,IT)) GO TO 2005
           WDOTHE(IF,IG,IT)=0.
           WHEADD = 0.0
           PRES(IP,IG,IT)=PTOT
           GO TO 2006
      2005 PHE=SOPRES(IG,IT)-PPV
      C                    CALC. RHO OF GAS
           CALL GSZDNS (I7,THE,PHE,RHOGT,Z)
           WH = RHOGT*PVOL
           WDOTHE(IF,IG,IT)=(WH-WHE)/(DCYCLE(IP-1))
           PRES(IP,IG,IT)=SOPRES(IG,IT)
           WHEADD = WH - WHE
           WHE=WH
      2006 PRESHE(IP,IG,IT)=PHE
           WHESUM = WHESUM + WHEADD
      C
           IF(DIAG(2,6HTANK6 )) WRITE (6,6080) IG,IT,IP,IW,IF,
          1                        T,Z,RHOG,RHOLI,PVOL,VLIQ,WP,WPV,
```

```
     2                         SOPRES(IG,IT),PPV,THE,Z1,WH,WHE,
     3                         DCYCLE(IP-1),WDOTHE(IF,IG,IT),
     4                         PRES(IP,IG,IT),PRESHE(IP,IG,IT)
C
      WRITE (IOT,7011)
      WRITE (IOT,7012) T, THE, PVOL, VLIQ, WP, WPV, PRESHE(IP,IG,IT),
     1           PTOT, SOPRES(IG,IT), WDOTHE(IF,IG,IT), WHEADD,
     2                 PRES(IP,IG,IT), WHESUM
C
      GO TO 90
C
C          ***** DO FINAL CALCULATIONS FOR SYSTEM 2.
C
C          ***** CALCULATE PRESSURIZATION SYSTEM WEIGHT.
C
 2010 SUMWDH = 0.0
      DO 2020 II = 1,IF
      SUMWDH = SUMWDH +WDOTHE(II,IG,IT)*DCYCLE(2*II-1)
 2020 CONTINUE
C
      WHETOT(IG,IT) = WHESUM
C
      WPGTOT(IG,IT) = 1.5*WHETOT(IG,IT) + 40.0
      IF(DIAG(2,6HTANK8 )) WRITE(6,6100) IG,IT,IP,IW,IF,SUMWDH,
     1                                                 WHETOT(IG,IT),
     2              WPGTOT(IG,IT), WPTOT(IG), PWTOT(IG)
C
      WRITE (IOT,7015)
      WRITE (IOT,7016) WHETOT(IG,IT), WPGTOT(IG,IT)
C
      GO TO 30
C
C          ***** SYSTEM 3 *****
C
C
C          ***** CALCULATE PRESSURANT GAS FLOW RATE.
C
 3000 T     = TSAT(SOPRES(IG,IT),IG)
      IF = IF + 1
C                    CALC. RHO OF GAS
      CALL GSZDNS (IG,T,SOPRES(IG,1),RHOG,Z)
      CALL RHOLIQ(T,IG,RHOL1)
      PVOL=SVOL(IG,IT)*(RHOL1-RHOP)/(RHOL1-RHOG)
      PRES(IP,IG,IT)=SOPRES(IG,IT)
      PPV=PRES(IP,IG,IT)
      WPVG=PVOL*RHOG
      WDOTPG(IF,IG,IT)=(WPVG-WPV)/(DCYCLE(IP-1))
      WPV=WPVG
      IF(DIAG(2,6HTANK9 )) WRITE (6,6012) IG,IT,IP,IW,IF,
     1                         T,Z,RHOG,RHOL1,PVOL,PRES(IP,IG,IT),PPV,
     2                         WPVG,WDOTPG(IF,IG,IT),DCYCLE(IP-1),WPV
C
      GO TO 90
C
C          ***** DO FINAL CALCULATIONS FOR SYSTEM 3.
C
C          ***** FIND MAXIMUM FLOWRATE OF THE PROPELLANT GAS.
C                * LOOK UP THE HEX WEIGHT PER FLOWRATE FOR THE HEX DELTA P.
```

********* TANK *********

```
      C          ***** CALCULATE THE WEIGHT OF THE HEX.
      C
      C
      C      DELETE NON-EXISTANT TABLES
      C
       3010 CONTINUE
      C
            WDPGMX = WDOTPG(1,IG,IT)
            DO 3020 II = 2,IF
            WDPGMX = AMAX1(WDPGMX,WDOTPG(II,IG,IT))
       3020 CONTINUE
      C
            JX = JX + 1
      C
            WDOTX(JX,IG) = WDPGMX
            UCODE(JX,IG) = HXCODE(JX,IG)
      C
            CALL HEATEX(IG  ,JX,WDOTX(JX,IG  ),HEXHIT(JX,IG  ),HEXCIT(JX,IG  )
           1,HEXHOT(JX,IG  ),HEXCOT(JX,IG  ),HEXHIP(JX,IG  ),HEXCIP(JX,IG  ),
           2 HEXHOP(JX,IG  ),HEXCOP(JX,IG  ),HXMRAT(JX,IG  ),WDOTH (JX,IG  ),
           3 WHXTOT(JX,IG  ))
      C
            WTHXPG(IG) = WHXTOT(JX,IG)
      C
      C          ***** COMPUTE THE WEIGHT OF GAS GENERATOR PROPELLANT REQUIRED
      C
            CALL FINTAB (NTBID(31),IG)
            XTAB(2) = SPGTEM(IG,IT)
            XTAB(3) = SHOTEM(IG,IT)
            WT = 0.0
            DO 3030 II = 1,IF
            XTAB(1) = PRES(2*II-1,IG,IT) + SPDELP(IG,IT)
            XTAB(4) = WDOTPG(II,IG,IT)
            WT = WT + MIPE(4,XTAB)*DCYCLE(2*II+1)
       3030 CONTINUE
      C
            WGGPPG(IG,IT) = WT
      C
      C          ***** LOOK UP GAS GENERATOR SYSTEM WEIGHT.
      C
            ATERM = 13.824204 - (0.01117823*SGGPC( IG, IT )) + (1.8632927E-5 *
           1(SGGPC( IG, IT )**2)) - (1.108423E-8 * (SGGPC( IG,IT  )**3))
      C
            BTERM = 7.9470262 - (.035636198*SGGPC( IG,IT  )) + (6.4684644E-5 *
           1(SGGPC( IG,IT  )**2)) - (3.7946E-8 * (SGGPC( IG,IT  )**3))
      C
            WGGAPG(IG,IT) = ATERM + BTERM * WDPGMX
      C
      C          ***** CALCULATE MOTOR HOURE POWER REQUIRED.
      C          ***** NOTE - OVERALL MOTOR PUMP EFFECIENCY FIXED AT 0.5.
      C
            CALL RHOLIQ(SITEMP(IG,IT),IG,R)
            HPMXPG = 144.0*SPDELP(IG,IT)*WDPGMX/(550.0*0.5*R)
      C
      C          ***** LOOK UP MOTOR WEIGHT. MOTOR SPEED = CONSTANT = 30000.0
      C
            CALL FINTAB (NTBID(33))
            XTAB(1) = HPMXPG
```

********   TANK   ********

```
      XTAB(2) = 30000.0
      WMPG(IG,IT) =           MIPE(2,XTAB)
C
C         ***** CALCULATE BATTERY WEIGHT.
C
      WB = 0.0
      CNST = 144.0*SPDELP(IG,IT)*746.0/(550.0*0.5*R*3600.0)
C
      DO 3040 II = 1,IF
      WB = WB + CNST*WDOTPG(II,IG,IT)*DCYCLE(2*II+1)
C
 3040 CONTINUE
C
      WBPG(IG,IT) =  WB           /PDNSTY
C
C         ***** LOOK UP PUMP WEIGHT.
C         ***** NOTE - DELTA PRES. NPSH = 0, AND EFFICIENCY = .7
C
      CALL FINTAB (NTBID(34)+IG)
      XTAB(1) = .7
      XTAB(2) = 0.0
      XTAB(3) = SPDELP(IG,IT)
      XTAB(4) = WDPGMX
      WCPPG(IG,IT) =           MIPE(4,XTAB)
C
C         ***** DETERMINE THE PRESSURIZATION SYSTEM WEIGHT.
C
      WTSYPG(IG,IT) = WTHXPG(IG) + WGGAPG(IG,IT) + WMPG(IG,IT) +
     1                WBPG(IG,IT) + WCPPG(IG,IT)
C
C         ***** SEPARATE THE GAS GENERATOR PROPELLANT WEIGHT INTO
C         ***** H2 AND O2 COMPONENTS
C
      WTGGH2(IG,IT) = WGGPPG(IG,IT)/(SGMRAT(IG,IT) - 1.0)
      WTGGO2(IG,IT) = WGGPPG(IG,IT) - WTGGH2(IG,IT)
C
      IF(DIAG(2,6HTANK12)) WRITE (6,6015) IG,IT,IP,IW,IF,
     1    WDPGMX,WTHXPG(IG),WT,WGGPPG(IG,IT),WGGAPG(IG,IT),R,HPMXPG,
     2    WMPG(IG,IT),CNST,WB,WBPG(IG,IT),WCPPG(IG,IT),SPDELP(IG,IT),
     3    WTSYPG(IG,IT),WTGGH2(IG,IT),SGMRAT(IG,IT),WTGGO2(IG,IT)
C
      GO TO 30
C
 6000 FORMAT ('+'14XII5 / (15X4E15.8))
C
 6005 FORMAT('+',14X,T23,'-TEMP-',T37,'-WAPS-',T52,'-WP-',T67,'-RHOLI-',
     1 T82,'-VLIQ-',T97,'TANK-VOL'/15X,6E15.8/T23,'ULL-VOL',T37,'-RHOG-'
     2,T52,'-PPV-',T67,'-WTOT-'/15X,4E15.8)
C
 6010 FORMAT('+',14X,5I5/(15X,7E15.8))
C
 6012 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3/15X,T21,'TEMP-GAS',T37,'Z-GAS',T52,'RHOG',T67,'RHOLIQ',
     2 T81,'ULL-VOL',T97,'PRESSURE',T112,'-PPV-'/15X,7E15.8/15X,T23,
     3 'WPVG',T37,'WDOTPG',T52,'DCYCLE',T67,'-WPV-'/15X,4E15.8)
C
 6015 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3/15X,T23,'WDPGMX',T37,'WTHXPG',T52,'- WT -',T67,'WGGPPG',
```

```
     2 T82,'WGGAPG',T97,'- R -',T112,'HPMXPG'/15X,7E15.8/15X,T23,'WMPG',
     3 T37,'CNST',T52,'- WB -',T67,'WBPG',T82,'WCPPG',T97,'SPDELP'/15X,
     4 6E15.8/15X,T23,'WTSYPG',T37,'WTGGH2',T52,'SGMRAT',T67,'WTGGO2',
     5 /15X,4E15.8)
C
6020 FORMAT(/20X,20('*'),' BURN NUMBER =',I4,5X,'PRESS.SYS.NO. =',I4,
     1 1X,20('*')/)
C
6030 FORMAT(/20X,20('*'),' COAST NUMBER =',I4,5X,'PRESS.SYS.NO. =',I4,
     1 1X,20('*')/)
C
6031 FORMAT(/18X,20('*'),'  FINAL ENGINE SHUTDOWN PROPELLANT TANK CONDI
     1TIONS  ',20('*')/)
C
6033 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3/15X,T23,'RHOP',T34,'INT.ENERGY',T54,'WP',T69,'WPV',T84,
     2 'WPT',T96,'PRESSURE'/15X,6E15.8)
6035 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3/15X,T23,'SVWT',T37,'WTOT',T52,'PPV',T68,'WP',T83,'PHE',
     2  T96,'PRESSURE'/15X,6E15.8/15X,T23,'RHOP',T37,'RATIO',T51,
     3  'INT.ENERGY'/15X,3E15.8)
6040 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3,              /15X,T21,'ENTHALPY',T36,'WDOTJ',T51,'PWTOT',
     2 T66,'WTTOT',T81,'WDOTJM',T94,'INT.ENERGY',T113,'WTOT'/15X,7E15.8,
     3 /15X,T22,'DCYCLE',T38,'-E-'/15X,2E15.8)
6050 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3/15X,T23,'WOUT',T37,'WTOT',T53,'WPT',T66,'INT.ENERGY',
     2 T82,'WTTOT',T97,'PWTOT',T112,'-E-'/15X,7E15.8/15X,T23,'SVOL',
     3 T37,'RHOP'/15X,2E15.8)
6060 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3/15X,T23,'RHOP',T34,'INT.ENERGY',T54,'CVP',T69,'CVH',T82,
     2 'RATIO',T99,'PPV'/15X,6E15.8/T24,'PHE',T36,'PRESSURE',T54,'WP',
     3 T69,'WPV',T82,'RHOG'/15X,5E15.8)
6061 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3,/15X,T23,'RHOP',T34,'INT.ENERGY',T51,'RATIO',T70,'PPV',
     2 T80,'PRESSURE'/15X,5E15.8)
6070 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3,/15X,T21,'WVSUM',T36,'TFINAL',T53,'-Z-',T66,'SWVTOT',
     2  T83,'WGR',T97,'WLRT'/15X,6E15.8)
6080 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3,/15X,T22,'TSAT',T37,'Z-GAS',T51,'RHO-GAS',T66,'RHO-LIQ',
     2 T81,'ULL-VOL',T95,'LIQ-VOL',T112,'-WP-'/15X,7E15.8/T23,'-WPV-',
     3 T34,'OP.PRESSURE',T52,'PPV',T67,'HE-TEMP',T81,'Z-HELIUM',T97,
     4 '-WH-',T112,'-WHE-'/15X,7E15.8/T23,'DCYCLE',T37,'WDOTHE',T52,
     5 'PRESSURE',T65,'HE-PRESSURE'/15x,4E15.8)
6090 FORMAT('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
     1 'IF=',I3,/15X,T22,'SVWT',T38,'- WP =',T53,'RHO',T64,'INT.ENERGY'/
     2 15X,5E15.8)
6100 FORMAT('+',14x,'IG=',I3,2x,'IT=',I3,2x,'IP=',I3,2x,'IW=',I3,2x,
     1 'IF=',I3/15X,T22,'SUMWDH',T38,'WHETOT',T51,'WPGTOT',T67,'WPTOT',
     2 T83,'PWTOT'/15X,5E15.8)
C
7000 FORMAT(/T42, '*** TANK AND VENT PARAMETER CALCULATIONS ***'/)
7001 FORMAT(T20,'*** INITIAL TANK CONDITIONS ***'/)
7002 FORMAT(T5,'FLUID CONSIDERED - ',2A6,T43,'FLUID TEMPERATURE      = '
     1,F8.2,T81,'TANK INITIAL PRESSURE = ',F8.2/T5,'WGT.OF LIQ. PROP.
     2 = ',F8.2,T43,'WGT. PROP, VAPOR       = ',F8.3,
     3                                          T81,'WGT. LIQ. + VAPOR
```

********   TANK   ********

```
     4  = ',F8.2/T5,'WGT. HELIUM IN VAPOR = ',F8.2,T43,'TOTAL FLUIDS IN
     5TANK  = ',F8.2,T81,'VOL. OF LIQUID FLUID  = ',F8.2/T5,'PART.PRES.P
     6ROP.VAPOR  =',F8.3,T43,'PART.PRES.HELIUM GAS  = ',F8.3,T81,'ULLAGE
     7 VOLUME IN TANK = ',F8.2/T5,'TANK VOLUME              = ',F8.2,T43,'E
     8FF. TANK DENSITY      = ',F8.3,T81,'EFF. INTERNAL ENERGY  = ',E15.8
     9)
7003 FORMAT(T20,'*** PRE- OR NON-VENT CONDITIONS ***'/)
7004 FORMAT(T5,'FLUID CONSIDERED - ',2A6,T43,'FLUID TEMPERATURE     = '
     1,F8.2,T81,'COAST DURATION - SEC. = ',F8.0/T5,'WGT.OF LIQ. PROP.
     2  = ',F8.3,T43,'WGT. PROP. VAPOR        = ',F8.3,T81,'WGT.HELIUM IN
     3VAPOR  = ',F8.3/T5,'PART.PRES.PROP.VAPOR  = ',F8.3,T43,'PART.PRES.
     4HELIUM GAS  = ',F8.3,T81,'CURRENT TANK PRESSURE = ',F8.3/T5,'EFF.I
     5NTERNAL ENERGY   = ',E15.8)
7005 FORMAT(/T20,'*** POST VENT CONDITIONS ***'/)
7006 FORMAT(T5,'TANK VENT PRESSURE      = ',F8.2,T43,'WGT.VENTED FLUIDS
     1   = ',F8.2,T81,'WGT.OF LIQ.IN TANK    =',F8.2/T5,'WGT.VAPOR IN TA
     2NK      = ',F8.3,T43,'WGT.HELIUM IN VAPOR  = ',F8.3,T81,'TOTAL FLUI
     3DS IN TANK  = ',F8.2/T5,'PART.PRES.PROP.VAPOR  = ',F8.3,T43,'PART.
     4PRES.HELIUM GAS  = ',F8.3,T81,'VENTED TANK PRESSURE  = ',F8.3/T5,
     5 'EFF.INTERNAL ENERGY   = ',E15.8)
7007 FORMAT(T20,'*** COMPUTE ENERGY BALANCE FOR BURN ***'/)
7008 FORMAT(T5,'FLUID CONSIDERED - ',2A6,T43,'BURN DURATION - SEC.  = '
     1,F8.0,T81,'FLOWRATE FOR THRUST   = ',F8.3/T5,'THRUST PROP.REMAININ
     2G = ',F8.2,T43,'PROPELLANT IN TANK     = ',F8.2,T81,'EFF. INTERNAL
     3ENERGY  = ',E15.8/T5,'EFF. TANK ENERGY       = ',E15.8,
     4 T81,'TOTAL FLOWRATE          = ',F8.3)
7009 FORMAT(/T20,'*** COMPUTE RESULTING TANK CONDITIONS ***'/)
7010 FORMAT(T5,'PROPELLANT WITHDRAWN  = ',F8.3,T43,'TOTAL FLUIDS IN TAN
     1K  = ',F8.2,T81,'PROPELLANT LIQ.+VAP.  = ',F8.2/T5,'THRUST PROP.RE
     2MAINING = ',F8.2,T43,'NEW EFF. TANK DENSITY = ',F8.4,T81,'PART.PRE
     3S.PROP.VAPOR  = ',F8.3/T5,'NEW INTERNAL ENERGY   = ',E15.8)
7011 FORMAT(/T20,'*** COMPUTE PRESSURANT NEEDED FOR THIS BURN ***'/)
7012 FORMAT(T5,'TANK LIQ. TEMPERATURE = ',F8.2,T43,'STORED HELIUM TEMP.
     1   = ',F8.2,T81,'NEW TANK ULLAGE VOL.  = ',F8.3/T5,'NEW PROP. LIQ.
     2 VOLUME = ',F8.2,T43,'PROP. LIQ. REMAINING  = ',F8.2,T81,'WGT. OF
     3PROP. VAPOR    = ',F8.4/T5,'HELIUM PART.PRESSURE  = ',F8.3,T43,'TOT
     4AL PRES. *PPV+PHE* = ',F8.3,T81,'NOM. OPERATING PRES.  = ',F8.3/T5
     5,'HELIUM FLOW  RATE      = ',E10.4,T43,'WEIGHT OF HELIUM USED = ',
     6 E10.4,T81,'NEW TANK PRESSURE       = ',F8.3/T5,'TOTAL HELIUM CONSUM
     7ED = ',F8.3)
7013 FORMAT(T20, '*** COMPUTE FINAL TANK CONDITIONS ***'/)
7014 FORMAT(/T5,'FINAL TANK TEMP.       = ',F8.3,T43,'TOTAL VENTED GAS W
     1GT. = ',F8.3,T81,'WGT. OF GAS RESIDUALS = ',F8.3/T5,'WGT. OF LIQ.R
     2ESIDUALS = ',F8.3)
7015 FORMAT(/T20,'*** COMPUTE PRESSURIZATION SYSTEM WEIGHT ***'/)
7016 FORMAT(T5,'TOTAL HELIUM GAS REQD = ',F8.3,T43,'WGT.PRESSURANT SYST
     1EM = ',F8.3)
7017 FORMAT(/T42,'*** TANK AND VENT PARAMETER CALCULATIONS - CONTD. ***
     1'/)
C
C
      END
```

********* PRØCEDURE DEFINITIØN PRØCESSØR – TANKWT

```
TANKWT*  PROC
C
        PARAMETER N1W=10,N2W=2
C
        COMMON /TANKWT/ IWOP,NOSHAP,JTKTYP(N1W),JFLTP(N1W),XD(N1W),YD(N1W)
       1 ,ZD(N1W),TAR(N1W),TVL(N1W),THD(N2W),TOTVL(N2W)
C
        DIMENSION A1(N2W),A2(N2W),A3(N2W),V1(N2W),V2(N2W),V3(N2W),VMX(N2W)
C
        EQUIVALENCE (TOTVL,VMX),(TAR,A1),(TAR(3),A2),(TAR(5),A3),(TVL,V1),
       1 (TVL(3),V2),(TVL(5),V3)
C
 END
```

********* FUNCTIØN TBØIL

```
      FUNCTION TBOIL(I)
      DIMENSION TC(17)
      DATA(TC(N),N=1,17)/277.85,59.8,259.13,59.8,260.,343.2,387.2,
     1 343.2,387.2,521.8,259.13,730.,776.4,1094.,749.5,1155.,9.37/
      TBOIL=TC(I)
      RETURN
      END
```

******** SUBRØUTINE TCØND

```
      SUBROUTINE TCOND(TH,TC,NBAR,THKIN,INTYPE,QCOND)
C     *
C     THIS SUBROUTINE COMPUTES THE THERMAL CONDUCTIVITY FOR NINE
C     INSULATION MATERIALS.  REF., LMSC-A964947-VOL.II, LMSC A981608,
C     WITH MICROSPHERE EQNS. PER  R.PARMLEY, EMITTANCE VALUES ARE FROM
C     LMSC A903316 (NASA CR-72605).
C     *
      REAL NSHLD,NBAR
C     *
      DELT = TH-TC
      TMEAN = (TH+TC)/2.0
      SUMT = TH+TC
      SUMSQT = (TH**2)+(TC**2)
      TMPR1 = TC/TH
      TMPR2 = TMPR1*TMPR1
      TH3 = TH*TH*TH
      NSHLD = THKIN*NBAR
      THKFT = THKIN/12.0
      THETA1 = (1.0+TMPR1)
      THETA2 = (1.0+TMPR2)
      SIGMA = 0.1713E-08
C     *
      GO TO (10,20,30,40,50,60,70,80,90),INTYPE
C     *
C     * FOR DOUBLE ALUMINIZED MYLAR-SILK NET
C     *
   10 SCNST = 2.22E-09
      EMIT1 = 4.40E-04 * (TMEAN**0.667)
      DEMIT = ((2.0/EMIT1)-1.0)
      GO TO 22
C     *
C     * FOR DOUBLE GOLDIZED MYLAR-SILK NET
C     *
   20 SCNST = 2.22E-09
      EMIT2 = 8.76E-04 * (TMEAN**0.509)
      DEMIT = ((2.0/EMIT2)-1.0)
   22 SCOND = SCNST*NBAR*TMEAN
      RNUM = SIGMA*SUMSQT*SUMT*THKFT
      RDEN = (NSHLD-1.0) * DEMIT
      RCOND = RNUM/RDEN
      QCOND = (SCOND+RCOND) *(DELT/THKFT)
      RETURN
C     *
C     * FOR DOUBLE ALUMINIZED MYLAR-TISSUE GLASS
C     *
   30 SCNST = 7.00E-12
      EMIT1 = 4.40E-04 * (TMEAN**0.667)
      DEMIT = ((2.0/EMIT1)-1.0)
      GO TO 42
C     *
C     * FOR CRINKLED DOUBLE ALUMINIZED MYLAR-TISSUE GLASS
C     *
   40 SCNST = 8.80E-12
      EMIT1 = 4.90E-04 * (TMEAN**0.67)
      DEMIT = ((2.0/EMIT1)-1.0)
   42 SCOND = SCNST*(NBAR**2)*TMEAN
      RNUM = 1.7*SIGMA*SUMSQT*SUMT*THKFT
```

```
********    TCOND    ********
      RDEN = (NSHLD-1.0) * DEMIT
      RCOND = RNUM/RDEN
      QCOND = (SCOND+RCOND) * (DELT/THKFT)
      RETURN
C     *
C     * FOR NRC-2 CRINKLED SINGLE ALUMINIZED MYLAR
C     *
   50 SCNST = 2.00E-10
      EMITA = 4.90E-04 * (TMEAN**0.67)
      EMITB = 5.58E-03 * (TMEAN**0.667)
      GO TO 62
C     *
C     * FOR SUPERFLOC
C     *
   60 SCNST = 15.40E-11
      EMITA = 4.40E-04 * (TMEAN**0.667)
      EMITB = 4.10E-01
   62 DEMIT = ((1.0/EMITA)+(1.0/EMITB)-1.0)
      SCOND = SCNST*(NBAR**2)*TMEAN
      RNUM = SIGMA*SUMSQT*SUMT*THKFT
      RDEN = (NSHLD-1.0)*DEMIT
      RCOND = RNUM/RDEN
      QCOND = (SCOND+RCOND)*(DELT/THKFT)
      RETURN
C     *
C     *  FOR MICROSPHERES (104 TO 135 MICRONS)
C     *
   70 RCNST = 1.56E-13
      RCOND = RCNST*TH3*THETA1*THETA2
      QCOND = RCOND * (DELT/THKFT)
      RETURN
C     *
C     * FOR POLYURETHANE FOAM
C     *
   80 PKSUBE = 1.1295E-03 + (3.481E-05 * TMEAN)
      QCOND = PKSUBE * (DELT/THKFT)
      RETURN
C     *
C     * FOR FIBERGLASS BATTING - HELIUM PURGED
C     *
   90 FKSUBE = 1.3836E-03 * (TMEAN**0.662)
      QCOND = FKSUBE * (DELT/THKFT)
      RETURN
C     *
      END
```

********* SUBRØUTINE TEL

```
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C                    * ROUTINE NAME - TABLE EVALUATION OR LOOKUP  *
C                    * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                    * PROGRAMMER   - R. BOLLINGER 1949 102 26933 *
C                    * DATE CODED   - 2/25/70                     *
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE TEL(X,Y)
C
C         ***** EXPLANATION OF THE CALLING SEQUENCE
C
C                X - VALUE OF THE INDEPENDENT VARIABLE TO BE
C                    INTERPOLATED ON.
C
C                Y - VALUE OF THE DEPENDENT VARIABLE WHICH IS THE
C                    INTERPOLATION RESULTS.
C
      INCLUDE CIOUNT
      INCLUDE CKEYS
      INCLUDE CTAB
C
 6000 FORMAT ('0'10X'*** INTERPOLATION ERROR - OVERFLOW OR NV WRONG NV='
     1  I3,' ***')
 6100 FORMAT (1H0 10X'***  INDEP. VAR. =' F10.4, ' OUT OF TABLE RANGE ('
     1  F10.4, ' TO' F10.4, ' ) TABLE NO.'I4,' NO. SUBTABLES'I3,'  ***')
C
C
      IF(TYPE.EQ.1) GO TO 30
C
C         ***** EVALUATE THE COEFICIENT POLYNOMIAL FOR Y.
C
      Y = XTAB(1)
      DO 20 II = 2,NV
      Y = XTAB(II) + X*Y
   20 CONTINUE
      RETURN
C
C         ***** EVALUATE THE TABLE LOOKUP FOR Y.
C
   30 IF (XTAB(1) .LE. XTAB(NV)) GO TO 33
C                                  MAKE INDEPENDENT VAR. NON-DECREASING
      NV1 = NV + 1
      NVD2 = NV / 2
      DO 32 I=1,NVD2
      PIVOT = XTAB(NV1-I)
      XTAB(NV1-I) = XTAB(I)
      XTAB(I) = PIVOT
      PIVOT = YTAB(NV1-I)
      YTAB(NV1-I) = YTAB(I)
   32 YTAB(I) = PIVOT
   33 IF (KEY1 .LT. 1612) GO TO 34
C                                  TEST FOR INDEP. VAR. OUT OF RANGE
      IF (X .GE. XTAB(1) .AND. X .LE. XTAB(NV)) GO TO 34
      WRITE (IOT,6100) X,XTAB(1),XTAB(NV),JTABID,NLTBL
      IF (KEY1 .NE. 1612) GO TO 34
      KEY2 = KEY1
      RETURN
```

```
********     TEL     *********

        34 Y = YLGINT(XTAB,YTAB,NV,X,NIP,$40)
           RETURN
        40 WRITE (IOT,6000) NV
           CALL EXIT
     C
           END
```

******** SUBRØUTINE TEMP

```
      SUBROUTINE TEMP(T)
C
C.... ROUTINE TO CONVERT NBS-55 TO IPTS-68 IN THE RANGE BELOW NBP-OXYGEN
C.... (90.18 NBS-55 OR 90.188 IPTS-68) BY INTERPOLATION FROM TABULATED VALUES
C.... GIVEN IN METROLOGIA 5, 47(1967)
C
C.... ROUTINE TO CONVERT IPTS-48 TO IPTS-68 BY METHOD GIVEN BY
C.... THOMAS B. DOUGLAS (JOURNAL OF RESEARCH NBS VOL 73A NO5 SEPT-OCT 1969)
C.... IN THE RANGE 90.18 TO 10000 K.
C
      DIMENSION CHNG(38)
C
      DATA (CHNG(I),I=1,38) /8.900,7.100,5.200,3.400,1.700,0.300,-0.800,
     2 -1.400,-1.500,-1.200,-0.700,-0.100,0.500,0.900,1.100,0.900,0.300,
     3 -0.600,-1.700,-3.000,-4.300,-5.600,-6.800,-7.800,-8.600,-9.000,
     4 -9.000,-8.600,-7.700,-6.400,-4.900,-2.900,-0.500,2.200,4.900,
     5 7.400,9.600,11.100/
      IF((T.LT.273.150100).AND.(T.GT.273.149900))RETURN
      IF(T.GT.90.1800)GO TO 2
      IT=T
      TI=IT
      IT=IT-53
      DELT=(CHNG(IT)+((CHNG(IT+1)-CHNG(IT))*(T-TI)))/1000.00
      T=T+DELT
      RETURN
    2 IF(T.GT.273.1500)GO TO 3
      C=T-273.15
      CALL WFIND(T,W)
      TOP=1.E0+3.984517E-3*C-5.855019E-7*C**2+4.35717E-12*(100.00-C)*
     1    C**3-W
      TOP=TOP*250.97
      BOT=1.00-2.9389E-4*C+4.3741E-9*(75.00-C)*C**2
      DELT=TOP/BOT
      T=T+DELT
      RETURN
    3 IF(T.GT.903.8900)GO TO 4
      C=T-273.15
      TOP=4.904E-7*C*(C-100.00)
      BOT=1.00-2.939E-4*C
      PT=0.045
      PT=PT*(C/100.00)
      PT=PT*(C/100.00-1.00)
      PT=PT*(C/419.5800-1.00)
      PT=PT*(C/630.7400-1.00)
      DELT=TOP/BOT
      DELT=DELT+PT
      T=T+DELT
      RETURN
    4 IF(T.GT.1337.5800)GO TO 5
      C=T-273.15
      TOP=-1.3145+1.5016E-3*C+1.5625E-6*C**2
      BOT=1.000+4.101E-4*C
      DELT=TOP/BOT
      T=T+DELT
      RETURN
    5 IF(T.GT.10273.1500)WRITE(6,100)T
  100 FORMAT(' TEMP TO BE CONVERTED EXEEDS 10000 C INPUT WAS',F20.10)
```

```
********    TEL    ********

         34 Y = YLGINT(XTAB,YTAB,NV,X,NIP,$40)
            RETURN
         40 WRITE (IOT,6000) NV
            CALL EXIT
      C
            END
```

********* TEMP *********

```
      E= EXP(-22135.00/T)
      DELT=5.56E-4*T+3.84E-7*(1.00-E)*T**2
      T = T + DELT
      RETURN
      END
```

********* SUBRØUTINE THETAB

```
      SUBROUTINE THETAB(DB,CPB,D1B,D2B,SHI)
C
C     CALCULATES SPECIFIC HEAT INPUT (THETA) FROM EQN OF STATE
C     IN BRITISH UNITS
C     MUST FOLLOW CALLS OF PROPB,CPVTDB,DPDDB, AND DPDTB TO DEFINE
C     DB,DPB,D1B,+D2B
C     INPUT AND OUTPUT PARAMETERS ARE IN BRITISH UNITS
      SHI=DB*CPB*(D1B/D2B)
      RETURN
      END
```

```
C
      SUBROUTINE THKWTG (KFLG,IGAS,ITYPE,MTYPE,TEMP,PU,HD,AREA,HT,RAD,
     1                   RAD1,TWEGHT,KMT)
C
C                 * * * * * * * * * * * * * * * * * * * * *
C                 * THIS ROUTINE CALCULATES WALL -        *
C                 * THICKNESS AND WEIGHT   (IF NOT        *
C                 *          SPECIFIED WEIGHT IS          *
C                 *          MINIMIZED)                   *
C                 * DATE CODED    JULY 1972               *
C                 * PROGRAMMER    J. MCKAY , D19-43       *
C                 *               BLDG 201   X45178       *
C                 * * * * * * * * * * * * * * * * * * * * *
C
      INCLUDE  CMATRL,LIST
      INCLUDE TABLOK
C
      DIMENSION ER(5),IM(3),RK(5,2),WT(3)
C
      DATA  IM/2,1,5/
      DATA  ER/1.,1.25,1.5,1.75,2./
      DATA (RK(I,1),I=1,5) / .50,.605,.80,.925,1.13 /
      DATA (RK(I,2),I=1,5) / .67,.745,.84,.955,1.20 /
C
C                 KFLG = 1 FOR COMPLETE ELLIPSOID
C                      = 2 ELLIPSOID CONNECTED TO OTHER SHAPE
C                 IGAS = 1 FOR O2   , = 2 FOR H2
C                 ITYPE= TANK SHAPE 1=CYL,2=FRUS.OF CONE,3=ELLIPSOID
C                          4=CYD.-ELLIPSOID
C                 MTYPE= TANK MATERIAL TYPE (IF NOT INPUT PROGRAM WILL
C                                                  SELECT LIGHTEST)
C
C                 FIND FLUID DENSITY
      CALL FDNSTY (IGAS,TEMP,PU,RHOF)
C                 TANK PRESSURE
      P = PU + 0.0229*HD*RHOF
      JKM = 0
      IF (P .GE. 1000.) JKM = 5
      IF (P .GE. 3000.) JKM = 10
      ISTP = 1
      IF (MTYPE .EQ. 0) ISTP = 3
      KMT = IM
C
      DO 200 I=1,ISTP
      IMTR = IM(I)
      IF (MTYPE .EQ. 0) GO TO 100
      KMT = MTYPE
      IMTR = MTYPE
  100 CALL FINTAB (NTBID(25)+IMTR)
C                 LOOKUP FTU FOR THE PARTICULAR MATERIAL (IMTR)
      FTU = MIPE(1,TEMP)
C                 CALCULATE THICKNESS FOR THE GIVEN SHAPE
      GO TO (110,120,130,110),ITYPE
C                 CYLINDER OR CYLINDER - ELLIPSOID
  110 THK = 2.0*P*RAD / FTU
      GO TO 180
C                 FRUSTRUM OF CONE
```

```
*********    THKWTG    *********
      C                         CALC. CONE HALF ANGLE
        120 COSALP = HT / (SQRT(HT*HT + (RADI-RAD)**2))
            THK = 2.0*P*AMAX1(RAD,RADI) / (COSALP*FTU)
            GO TO 180
      C                         ELLIPSOID (SPHERICAL)
      C                         CALC. ELLIPSE RATIO
        130 ERATO = AMAX1(HT,RAD)  /  AMIN1(HT,RAD)
      C                         INTERPOLATE ON ERATO TO FIND K
            DO 140 J=1,5
            JI = J
            IF (ERATO - ER(J)) 140,160,150
        140 CONTINUE
      C
        150 YK = RK(JI-1,KFLG)+(ERATO-ER(JI-1))*(RK(JI,KFLG)-RK(JI-1,KFLG))
           1                    / (ER(JI)-ER(JI-1))
            GO TO 170
        160 YK = RK(JI,KFLG)
        170 THK = (ERATO/2.+YK)*P*AMAX1(HT,RAD)  /  (2.0*FTU)
      C                         CONVERT THICKNESS FROM FEET TO INCHES
        180 THK = 12.0 * THK
      C                         CHECK FOR LESS THAN ALLOWABLE THICKNESS
            IF (THK .LT. MINTHK(IMTR+JKM)) THK = MINTHK(IMTR+JKM)
      C                         CALC. WEIGHT FOR EACH MATERIAL
            WT(I) = 0.1125 * RHOL(IMTR) * THK * AREA
      C                         SAVE MIN WEIGHT AND MATERIAL TYPE
            IF (WT(I) .GE. WT) GO TO 200
            WT = WT(I)
            KMT = IMTR
        200 CONTINUE
      C                         SET WEIGHT EQUAL TO LIGHTEST
            TWEGHT = WT
            RETURN
            END
```

********* FUNCTIØN VFUNC (TKGEØM)

```
C
      FUNCTION VFUNC (I)
C
C                     * * * * * * * * * * * * * * * * * * * *
C                     * ROUTINE INTERPRETS INPUT DIMENSIONS *
C                     * TO CALL GEOMETRY PROGRAMS           *
C                     * DATE CODED    JULY 1972             *
C                     * PROGRAMMER    J. MCKAY , D19-43     *
C                     *               BLDG 201   X45178     *
C                     * * * * * * * * * * * * * * * * * * * *
C
      INCLUDE  TANKWT,LIST
C
      JMP = IABS (JTKTYP(I))
C
C                     VOLUME CALCULATION ROUTINES
      GO TO (100,110,120,130),JMP
C                     CYLINDER
  100 VFUNC = CYLNDR (YD(I),XD(I))
      IF (IWOP .EQ. 3) VFUNC = 0.
      GO TO 500
C                     FRUSTRUM OF CONE
  110 VFUNC = FRCONE (YD(I),XD(I),ZD(I))
      GO TO 500
C                     ELLIPSOID
  120 VFUNC = HSPHER (XD(I),YD(I))
      GO TO 500
C                     CYLINDER LESS ELLIPSOID
  130 VFUNC = CYLSPH (XD(I),YD(I))
      GO TO 500
C
C                     AREA CALCULATION ROUTINES
      ENTRY AFUNC (I)
C
      JMP = IABS (JTKTYP(I))
      GO TO (200,210,220,230),JMP
C                     CYLINDER
  200 VFUNC = ARACYL (YD(I),XD(I))
      GO TO 250
C                     FRUSTRUM OF CONE
  210 VFUNC = AREAFR (YD(I),XD(I),ZD(I))
      GO TO 250
C                     ELLIPSOID
  220 VFUNC = ARSPHR (XD(I),YD(I))
      GO TO 250
C                     CYLINDRICAL + ELLIPSOIDAL SHAPE
  230 VFUNC = ARACYL (YD(I),XD(I))
C                     SHOULD AREA OF ELLIPSOID BE INCLUDED
      IF (JFLTP(I) .LT. 0) GO TO 500
C                     YES INCLUDE IT
      VFUNC = VFUNC + ARSPHR (XD(I),YD(I))
      GO TO 500
  250 IF (JFLTP(I) .LT. 0) VFUNC = 0.
      GO TO 500
C
C                     HEAD CALCULATION ROUTINES
      ENTRY HFUNC (I,PVOL)
```

```
C
      JMP = IABS (JTKTYP(I))
      GO TO (300,310,340,370),JMP
C                      CYLINDER
  300 CALL CYLHED (PVOL,YD(I),HD)
      VFUNC = HD
      GO TO 500
C                      FRUSTRUM OF CONE
  310 PV = PVOL
      IF (JTKTYP(I) .GT. 0) GO TO 320
C                      INVERTED FRUST. OF CONE
      PV = TVL(I) - PV
  320 CALL FRHEAD (PV,YD(I),ZD(I),XD(I),HD)
      IF (JTKTYP(I) .GT. 0) GO TO 330
      HD = XD(I) - HD
  330 VFUNC = HD
      GO TO 500
C                      ELLIPSOID
  340 PV = PVOL
      IF (JTKTYP(I) .GT. 0) GO TO 350
C                      INVERTED ELLIPSOID
      PV = TVL(I) - PV
  350 CALL ELIPSG (PV,XD(I),YD(I),HD)
      IF (JTKTYP(I) .GT. 0) GO TO 360
      HD = XD(I) - HD
  360 VFUNC = HD
      GO TO 500
C                      HEAD BETWEEN CYLINDER AND ELLIPSOID
  370 CALL CYMSPH (PVOL,XD(I),YD(I),HD)
      VFUNC = HD
  500 CONTINUE
      RETURN
      END
```

********* FUNCTIØN TMELT

```
      FUNCTION TMELT(P,K)
      IF(K.EQ.2)GO TO 1
      C    = 1.769
      PO   = 2637.2
      PT   = 0.00150
      TT   = 54.3507
      X    = 1.0/C
      T    = ((TT**C) * ((P/PO)-(PT/PO)+1.0))**X
      CALL TEMP(T)
      TMELT=T
      RETURN
    1 CONTINUE
      A    = -1379.0825
      B    = 0.9263018
      C    = 1.0/1.795
      T=((P-A)/B)**C
      TMELT=T
      RETURN
      END
```

********* FUNCTIØN TMELT B

```
      FUNCTION TMELTB(PB,K)
      P  = PB * 6.8947572E+3/1.01325E+5
      T=TMELT(P,K)
      TMELTB = T * 1.8
      RETURN
      END
```

******** SUBRØUTINE TNKWTA

```
C
      SUBROUTINE TNKWTA (ISW,FLDVOL,PCULLG,DIAM,TKTEMP,TKPRES,MTYPE,
     1                  TNKVL,WTOFTK,TOTARA,HC)
C
C                   * * * * * * * * * * * * * * * * * * *
C                   * ROUTINE CONTROLS THE CALCULATION OF *
C                   * TANK VOL., AREAS, HEAD, WALL THICK- *
C                   * NESS AND WEIGHTS.                   *
C                   * FOR BOTH OXYGEN AND HYDROGEN TANKS  *
C                   * DATE CODED     JULY 1972            *
C                   * PROGRAMMER     J. MCKAY , D19-43    *
C                   *                BLDG 201    X45178   *
C                   * * * * * * * * * * * * * * * * * * *
C
      INCLUDE CONST,LIST
      INCLUDE  TANKWT,LIST
C
      DIMENSION  DIAM(N2W),FLDVOL(N2W),MTYPE(N2W),PCULLG(N2W),RMAX(N2W),
     1  TKPRES(N2W),TKTEMP(N2W),TNKVL(N2W),TOTARA(N2W),WTOFTK(N2W)
     2 ,HC(N2W)
C
      DO 100 I=1,2
      RMAX(I) = DIAM(I) / 2.
      TNKVL(I) = FLDVOL(I) / (1.0-PCULLG(I)/100.)
  100 TOTVL(I) = 0.
C                   ISW = 1 STORAGE TANKS
C                   ISW = 2 ACCUMULATOR TANKS
      IF (ISW .EQ. 2) GO TO 110
      IF (IWOP .GT. 1) GO TO 200
C                   CALCULATE MAX. VOLUME  (SPHERE)
  110 DO 190 I=1,2
      VMX(I) = SPHERE (RMAX(I),RMAX(I))
      IF (VMX(I) .LE. TNKVL(I)) GO TO 130
C                   CALC. NEW RADIUS OF SPHERE
      RMAX(I) = (FLDVOL(I) / (2.*PI203))**(1./3.)
      DIAM(I) = 2.0*RMAX(I)
  120 V2(I) = 0.
      V1(I) = TNKVL(I) / 2.
      V3(I) = V1(I)
      HC(I) = 0.
      GO TO 140
  130 IF (VMX(I) .EQ. TNKVL(I)) GO TO 120
C                   REQUIRED VOL. GREATER THAN MAX SPHERE
C                   CALC. VOL OF ADDED CYLINDRICAL SECTION
      V2(I) = TNKVL(I) - VMX(I)
C                   CALC. HEIGHT OF CYLINDER
      HC(I) = V2(I) / (PI*RMAX(I)**2)
      V1(I) = VMX(I) / 2.
      V3(I) = V1(I)
  140 VMX(I) = TNKVL(I)
C                   CALCULATE HEAD FOR OPTION IWOP = 1
      IF (FLDVOL(I) .GT. V1(I)) GO TO 150
C                   HEAD IN LOWER HEMIS
      CALL SPHSEG (FLDVOL(I),RMAX(I),THD(I))
      GO TO 170
  150 THD(I) = RMAX(I)
      IF (FLDVOL(I) .GT. V1(I)+V2(I)) GO TO 160
```

```
C                    HEAD IN CYLINDER
      PVOL = FLDVOL(I)-V1(I)
      CALL CYLHED (PVOL,RMAX(I),HC1)
      THD(I) = THD(I) + HC1
      GO TO 170
  160 THD(I) = THD(I) + HC(I)
      PVOL = TNKVL(I) - FLDVOL(I)
C                    HEAD IN UPPER HEMIS
      CALL SPHSEG (PVOL,RMAX(I),HC1)
      THD(I) = THD(I) + RMAX(I) - HC1
C                    FIND AREAS
  170 A1(I) = ARSPHR (RMAX(I),RMAX(I))
      A3(I) = A1(I)
      A2(I) = ARACYL (RMAX(I),HC(I))
C                    TOTAL AREA
      TOTARA(I) = A1(I) + A2(I) + A3(I)
C                    CALCULATE TANK WEIGHT (IWOP=1)
      KFLG = 1
      WTG1 = 0.
      IF (HC(I) .EQ. 0.) GO TO 180
      KFLG = 2
C                    CALC. WEIGHT OF CYLINDRICAL SECTION (IF NECESSARY)
      CALL THKWTG (KFLG,I,1,MTYPE(I),TKTEMP(I),TKPRES(I),THD(I),A2(I),
     1  HC(I),RMAX(I),RMAX(I),WTG1,MFLG1)
C                    CALC. WEIGHT OF HEMISPHERICAL ENDS
  180 CALL THKWTG (KFLG,I,3,MTYPE(I),TKTEMP(I),TKPRES(I),THD(I),
     1      2.0*A1(I),RMAX(I),RMAX(I),HC,WTG2,MFLG1)
C
C                    TOTAL TANK WEIGHT (O2 OR H2)  (MAIN OR ACCUMULATORS)
      WTOFTK(I) = WTG1 + WTG2
  190 CONTINUE
      RETURN
C
C                    CALCULATIONS FOR GENERAL TANK SHAPE INPUT
C                    MAIN STORAGE TANKS ONLY  (IWOP = 2 OR 3)
C
  200 KF = 1
      ISV = NOSHAP + 1
C                    FIND FLUID TYPE OF BOTTOM TANK
      KFL = IABS(JFLTP)
      K1 = 1
      K2 = 2
      K3 = 1
      IF (KFL .EQ. 1) GO TO 210
      K1 = 2
      K2 = 1
      K3 = -1
C                    CALC. VOLUME AND AREA OF TANKS
  210 DO 240 I=1,NOSHAP
      KFL2 = IABS(JFLTP(I))
  220 IF (KFL .EQ. KFL2) GO TO 230
C                    CHANGE IN FLUID TYPE
      ISV = I
      KF = 2
      KFL = KFL2
C                    VOLUME FOR A PARTICULAR TANK SHAPE
  230 TVL(I) = VFUNC (I)
      TOTVL(KFL) = TOTVL(KFL) + TVL(I)
```

```
C                    AREA FOR A PARTICULAR TANK SHAPE
      TAR(I) = AFUNC (I)
  240 CONTINUE
C                    HOW MANY SETS OF TANKS  (1 OR 2)
      IF (KF .EQ. 2) GO TO 250
      K1 = KFL
      K2 = K1
      K3 = 1
  250 IF (IWOP .NE. 3) GO TO 300
C
C                    FIT CYLINDRICAL SECTION (IWOP = 3)
C
      IST = 1
      DO 280 K=K1,K2,K3
C                    K IS THE FLUID TYPE FLAG
      DO 260 I=IST,NOSHAP
C                    SEARCH LIST FOR CYLINDER (ONE MUST BE PRESENT AND
C                    THE HEIGHT SHOULD BE ZERO FOR IWOP = 3)
      JCYL = I
      IF (IABS(JTKTYP(I)) .EQ. 1) GO TO 270
  260 CONTINUE
C                    CYLINDER IS AT JCYL IN LIST
  270 TVL(JCYL) = TNKVL(K) - TOTVL(K)
C                    FIT CYLINDRICAL SECTION TO REQUIRED VOLUME
      XD(JCYL) = TVL(JCYL) / (PI*YD(JCYL)**2)
C                    RE-CALCULATE AREA FOR FITTED CYLINDER
      TAR(JCYL) = AFUNC (JCYL)
      TOTVL(K) = TNKVL(K)
      IST = JCYL + 1
  280 CONTINUE
      GO TO 320
C                    RE-CALCULATE PERCENT ULLAGE VOLUME (IWOP = 2 ONLY)
  300 DO 310 K=K1,K2,K3
C                    TOTVL CALCULATED FROM THE INPUT DIMENSIONS
      PCULLG(K) = ((TOTVL(K) - FLDVOL(K)) / TOTVL(K))*100.
  310 TNKVL(K) = TOTVL(K)
C
C                    NOW CALCULATE HEAD FOR EACH TANK SET
  320 IST = 1
      IND = ISV - 1
      DO 370 K=K1,K2,K3
      THD(K) = 0.
      TOTARA(K) = 0.
      PV = 0.
      WTG1 = 0.
C
      DO 340 I=IST,IND
      PV = PV + TVL(I)
      IF (PV .LT. FLDVOL(K)) GO TO 330
      PVOL = FLDVOL(K) - PV + TVL(I)
C                    CALCULATE HEAD FOR THIS TANK SHAPE
      THD(K) = THD(K) + HFUNC(I,PVOL)
      GO TO 350
  330 THD(K) = THD(K) + XD(I)
  340 CONTINUE
C                    NOW CALCULATE TANK WEIGHTS
  350 DO 360 I=IST,IND
      JTKTP = IABS(JTKTYP(I))
```

```
********     TNKWTA     ********

      C
            CALL THKWTG (2,K,JTKTP,MTYPE(K),TKTEMP(K),TKPRES(K),THD(K),TAR(I),
           1     XD(I),YD(I),ZD(I),WTG2,MFLG1)
      C
            WTG1 = WTG1 + WTG2
      C                       CALC. TOTAL AREA THIS (02 OR H2) TANK
            TOTARA(K) = TOTARA(K) + TAR(I)
        360 CONTINUE
      C                       TOTAL WEIGHT FOR THIS (02 OR H2) TANK
            WTOFTK(K) = WTG1
      C                       RESET INDEX FOR UPPER TANK
            IST = ISV
            IND = NOSHAP
        370 CONTINUE
            RETURN
            END
```

******** SUBRØUTINE TRAC

```
C
      SUBROUTINE TRAC (NAME)
C
  100 FORMAT (1H0 '** TRACE ** ' A6)
C
      WRITE (6,100) NAME
      RETURN
      END
```

******** FUNCTIØN TSAT

```
      FUNCTION TSAT(P,I)
      GO TO (1,2,3,2,3,6,7,6,7,10,3,12,13,14,15,16),I
    1 T = 1519./(12.04-ALOG(P))
      GO TO 50
    2 T = 225.74/(8.7137-ALOG(P)) - .95
      IF(P.GT.19.)T=252.11/(9.4288-ALOG(P)) - .4
      IF(T.GT.59.8) T = 59.8
      GO TO 50
    3 T = 1374./(11.63-ALOG(P))
      IF(P.GT.200.)T = 1763./(13.43-ALOG(P))
      GO TO 50
    6 T = 1839./(11.83-ALOG(P))
      GO TO 50
    7 T = 1050./(5.73-ALOG10(P))
      GO TO 50
   10 T = 3168.7/(12.3579-ALOG(P))
      GO TO 50
   12 T = 5090./(14.45-ALOG(P))
      GO TO 50
   13 T = 7348.3/(16.54098-ALOG(P))
      GO TO 50
   14 T = 6650./(13.4055-ALOG(P))
      GO TO 50
   15 T = 1.8*1197./(7.4837-ALOG10(P/.01934))
      GO TO 50
   16 T = 1.8*1996./(8.2875-ALOG10(P/.01934))
   50 TSAT=T
      RETURN
      END
```

********* FUNCTIØN TSAT

DATE

```
      FUNCTION TSAT(P,I)
      GO TO (1,2,3,2,3,6,7,6,7,10,3,12,13,14,15,16),I
    1 T = 1519./(12.04-ALOG(P))
      GO TO 50
    2 T = 225.74/(8.7137-ALOG(P)) - .95
      IF(P.GT.19.)T=252.11/(9.4288-ALOG(P)) - .4
      IF(T.GT.59.8) T = 59.8
      GO TO 50
    3 T = 1374./(11.63-ALOG(P))
      IF(P.GT.200.)T = 1763./(13.43-ALOG(P))
      GO TO 50
    6 T = 1839./(11.83-ALOG(P))
      GO TO 50
    7 T = 1050./(5.73-ALOG10(P))
      GO TO 50
   10 T = 3168.7/(12.3579-ALOG(P))
      GO TO 50
   12 T = 5090./(14.45-ALOG(P))
      GO TO 50
   13 T = 7348.3/(16.54098-ALOG(P))
      GO TO 50
   14 T = 6650./(13.4055-ALOG(P))
      GO TO 50
   15 T = 1.8*1197./(7.4837-ALOG10(P/.01934))
      GO TO 50
   16 T = 1.8*1996./(8.2875-ALOG10(P/.01934))
   50 TSAT=T
      RETURN
      END
```

******** FUNCTIØN TSATH

```
      FUNCTION TSATH(TEMP,HG,HL)
      DIMENSION R(19),TL(19),TG(19),TF(19)
      DATA R/1.022,2.0,4.0,8.0,14.0,25.0,43.0,69.0,99.0,128.0,151.0,
     1165.,176.0,182.0,185.0,186.5,187.25,187.46875,187.506/
      DATA                                                  TF/24.845,
     227.07,29.81,33.07,36.18,39.96,44.12,48.33,51.97,54.79,56.72,57.80,
     358.57,58.99,59.18,59.29,59.34,59.353,59.356/
      DATA                                         TG/60.31,65.11,70.59
     4,76.35,80.98,85.11,87.40,86.54,81.94,74.15,64.83,56.86,47.34,39.56
     5,33.46,28.34,22.31,18.66,16.55/
      DATA                              TL/-132.8,-129.13,-124.25,-117.79
     6,-110.86,-101.3,-89.04,-74.22,-58.58,-43.43,-30.07,-20.56,-11.13,
     7-4.27,1.17,5.54,10.83,14.29,16.36/
      T=TEMP
      IF(T.LT.24.845)T=24.845
      IF(T.GE.59.356)T=59.356
      DO 104 I=2,19
      IF(T-TF(I))102,101,104
  101 HL=TL(I)
      HG=TG(I)
      TSATH=R(I)
      RETURN
  102 D=TF(I)-TF(I-1)
      TRR=TF(I)-T
      TTR=T-TF(I-1)
      HL=(TL(I)*TTR+TL(I-1)*TRR)/D
      HG=(TG(I)*TTR+TG(I-1)*TRR)/D
      TSATH=(R (I)*TTR+R (I-1)*TRR)/D
      RETURN
  104 CONTINUE
      RETURN
      END
```

******** SUBRØUTINE TSIZEI

```
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C                    * ROUTINE NAME - INITIAL TANK SIZE ROUTINE      *
C                    * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                    * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                    * DATE CODED   - 4/28/70                        *
C                    * REVISED      - JULY  1972                     *
C                    * PROGRAMMER   - J. MCKAY  D1943  201  45178 *
C                    * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE TSIZEI(IFLG)
C
      LOGICAL DIAG
C
      INCLUDE CACCUM
      INCLUDE CCNFIG
      INCLUDE CENG
      INCLUDE CHEX
      INCLUDE  CIOUNT
      INCLUDE CMATRL
      INCLUDE CMOTOR
      INCLUDE CTANK
      INCLUDE CTURBN
C
      DIMENSION WGGHEX(2),WGGTP (2)
C
      IF (DIAG(0,6HTSIZEI)) WRITE (IOT,6000) FLDLOD,SITEMP,SPGTEM,
     1                      SOPRES,SHFLUX,WGGTOT,(HXMRAT(1,I),I=1,2),
     2                      GWEGHT,TMRATO,TIPWT,MIXRAT
C
C         ***** COMPUTE THE TOTAL PROPELLANT USE BY THE ENGINES, HEX
C         ***** AND GAS GENERATOR FOR EACH GAS (O2 AND H2)
C
      H21 = WGGTOT(1)/(1.0 + HXMRAT(1,1))
      H22 = WGGTOT(2)/(1.0 + HXMRAT(1,2))
      WGGHEX(2) = H21 + H22
      WGGHEX(1) = WGGTOT(1) - H21 + WGGTOT(2) - H22
C
      H21 = GWEGHT(1)/(1.0 + TMRATO(1))
      H22 = GWEGHT(2)/(1.0 + TMRATO(2))
      WGGTP(2) = H21 + H22
      WGGTP(1) = GWEGHT(1) - H21 + GWEGHT(2) - H22
      IF(IFLG.EQ.2) GO TO 70
C
      WETOT2 = TIPWT/(1.0 + MIXRAT)
      WTTOT(2) = WETOT2 + .03 * WETOT2
C
      WETOT1 = TIPWT - WTTOT(2)
      WTTOT(1) = WETOT1 + .03 * WETOT1
C
      GO TO 80
C
   70 WTTOT(2) = TIPWT/(1.0 + MIXRAT)
      WTTOT(1) = TIPWT - WTTOT(2)
C
   80 CONTINUE
C
      DO 6 II=1,2
```

```
********   TSIZEI   ********
      IF (IFLG .EQ. 2) GO TO 4
C                    FIRST PASS
      WPTOT(II) = WTTOT(II) + WGGTP(II) + WGGHEX(II)
      WTOTP(II) = WPTOT(II)
      GO TO 6
C                    SECOND PASS
    4 WPTOT(II) = WPTOT(II) + WGGTP(II) + WTHXPG(II) + WGGPPG(II,1)
     1              + WGR(II,1) + WLR(II) + WGRACC(II)
    6 CONTINUE
C
      IF (DIAG(2,6HTSIZEI)) WRITE (IOT,6000) WPTOT,WTTOT,WGGTP,WGGHEX
C
C          ***** COMPUTE THE VOLUME AND HEAT RATE FOR EACH TANK FOR
C          ***** EACH GAS WITHIN THE INPUT TANK CONFIGURATION.
C
C
      DO 30 II=1,2
      CALL FDNSTY (II,SITEMP(II,1),SIPRES(II,1),R)
C                    HAS AN INITIAL LOAD BEEN INPUT
      IF (FLDLOD(II)) 20,20,10
C                    INITIAL LOAD INPUT -- USE IT
   10 IF (IFLG .EQ. 2) GO TO 20
      WTFLUD = FLDLOD(II)
      GO TO 30
C                    PROPELLANT LOAD CALCULATED
   20 WTFLUD = WPTOT(II)
C                    CALCULATE VOLUME OF FLUID
   30 SVLFLD(II) = WTFLUD / (R*NOP(II,1))
C
C                    CALCULATE TANK WEIGHT AND AREA
      CALL TNKWTA (1,SVLFLD,SULGPC,SMDIAM,SPGTEM,SOPRES,SMTYPE,SVOL,
     1              TWT,TSA,TCYHT)
C
      DO 60 II=1,2
      IF (IFLG .EQ. 2) GO TO 40
C                    FIRST PASS
      SHRATE(II,1) = TSA(II,1)*SHFLUX(II,1) / 3600.
      GO TO 50
C                    SECOND PASS
   40 IDXRI = SITYPE(II,1)
      TIWT(II,1) = NOP(II,1)* TSA(II,1)*RHOI(IDXRI)*SITHIK(II,1) / 12.
   50 TWT(II,1) = NOP(II,1)*TWT(II,1)
   60 CONTINUE
C
C                    OUTPUT THE TANK SIZING DATA
C
      CALL OTPTSZ (IFLG)
C
      IF (IFLG .EQ. 1) RETURN
      DO 100 II=1,2
      IDX = INDXTK(II)
      WI(IDX) = TIWT(II,1)
  100 WEIGHT(IDX) = TWT(II,1)
      RETURN
C
 6000 FORMAT ('+'14X,6E15.6/(15X6E15.6))
C
```

******** TSIZEI ********

116 END

******** FUNCTIØN TSTART

```
      FUNCTION TSTART(N)
C
      DIMENSION T(17)
C
      DATA (T(I),I=1,17)/162.3 , 36.5 ,153.04, 36.5 ,154.  ,201.  ,
     1                   231.4 ,201.  ,231.4 ,325.  ,153.04,432.  ,
     2                   530.1 ,618.2 ,466.8 ,652.1 ,  7.668/
C
      TSTART = T(N)
      RETURN
      END
```

********* SUBRØUTINE TVP

```
      SUBROUTINE TVP(P,T)
      COMMON /CTEVP/GT(8) /CRPR/CR(3)
      COMMON /SCRH/ X(40)
C
C.... ROUTINE TO SOLVE VAPOR PRESSURE EQUATION ITERATIVELY FOR
C.... TEMPERATURE BY NEWTON'S METHOD
C
      TC=CR(3)
C
C.... USE TEMP EXPLICIT EQN FOR FIRST APPROX
C
      P2=P*P
      P3=P2*P
      P4=P3*P
      P5=P4*P
      P6=P5*P
      X(1) = ALOG(P)
      X(2) = 1.0
      X(3)=P
      X(4)=P2
      X(5)=P3
      X(5)=P3
      X(6)=P4
      X(7)=P5
      X(8)=P6
      T    = 0.0
      DO 1 I=1,8
    1 T=T+X(I)*GT(I)
      T    = 1.0/T
C
C.... T IS NOW FIRST EST OF T
C
      ITRMAX=25
      EPS  = 1.0E-7
      DO 2 ITER=1,ITRMAX
      PP=VPN(T)
      CALL DPDTVP(T,P,DPDT)
      DELTA=(P-PP)/DPDT
      T=T+DELTA
      IF(ABS(DELTA/T).LT.EPS) RETURN
    2 CONTINUE
      WRITE(6,300)P,T,DELTA
  300 FORMAT(' *** TVP DID NOT CONVERGE',/,
     .       ' P   =',G15.7,
     .       ' T   =',G15.7,
     .       ' DEL =',G15.7)
      RETURN
      END
```

********* SUBRØUTINE TVPB

```
      SUBROUTINE TVPB(PB,TB)
      P    = PB * 6.8947572E+3/1.01325E+5
      CALL TVP(P,T)
      TB   = T * 1.8
      RETURN
      END
```

******** SUBRØUTINE VARNAM

```
C
      SUBROUTINE VARNAM
C
      INCLUDE  CACCUM
      INCLUDE  CAPU
      INCLUDE  CCNFIG
      INCLUDE  CCNTRL
      INCLUDE  CDCYCL
      INCLUDE  CENG
      INCLUDE  CHEX
      INCLUDE  CSYSWT
      INCLUDE  CHSORC
      INCLUDE  CIOUNT
      INCLUDE  CKEYS
      INCLUDE  CMATRL
      INCLUDE  CMOTOR
      INCLUDE  CNAMES
      INCLUDE  CONST
      INCLUDE  CPAGE
      INCLUDE  CPUMP
      INCLUDE  CTAB
      INCLUDE  CTABA
      INCLUDE  CTANK
      INCLUDE  CTURBN
      INCLUDE  DUMMY
      INCLUDE  SPUMP
      INCLUDE  TANKWT
C
      RETURN
C
      END
```

```
      SUBROUTINE VENT(Q,MH,MPV,ML,T,PV,PI,V,IG,PPVF,RHOP)
C
      LOGICAL JP,DIAG
C
      REAL MH,MPV,ML,MU,MT,MHF,MPVF,MTOT
C
      INCLUDE TABLOK
C
C     THIS SUBROUTINE COMPUTES TOTAL VENT MASS DURING COAST
C     AS WELL AS REVISED VALUES OF LIQUID AND ULLAGE MASSES
C     AND TEMPERATURE FOR A MIXED FLUID SYSTEM
C
C     *** DEFINITION OF SYMBOLS ***
C
C     Q       TOTAL AMOUNT OF HEAT ADDED TO TANK (BTU)
C     MH      MASS OF HELIUM IN ULLAGE (LBM)
C     MPV     MASS OF PROPELLANT VAPOR IN ULLAGE (LBM)
C     ML      MASS OF LIQUID IN TANK (LBM)
C     T       TEMPERATURE OF FLUID (LIQUID + VAPOR) IN TANK (R)
C     PV      VENT PRESSURE (PSIA)
C     PI      INITIAL PRESSURE (PSIA)
C     V       TOTAL TANK VOLUME (CU FT)
C     IG      PROPELLANT FLAG (1=OXYGEN, 2=HYDROGEN)
C
C     *****  DIAG SWITCH IS PLACED HERE  ******
C
      IF(DIAG(0,6HVENT  )) WRITE (6,6000) Q,MH,MPV,ML,T,PV,PI,V,IG,PPVF
C
C     *** INITIALIZE PRESSURE AND VENT MASS INCREMENTS ***
C
      UHI=0.
      UHF=0.
      DPM=1.
      DVTEST=1.0E-10
      DEP=PI-PV
      DP=DEP/10.
      IF(DP.GT.DPM) GO TO 95
      DP=AMIN1(DEP,DPM)
95    GO TO (100,110) ,IG
100   CMWR=7.996
      GC=48.25
      GO TO 120
110   CMWR=0.50365
      GC=766.
C
C     *** ENTER LOOP ON PRESSURE INCREMENT ***
C
120   DO 70 I=1,100
      LOOP=0
      DVMAX=0.99*(MPV+MH)
      DMVMAX=DVMAX
      DMVMIN=0.000000001
C     *** CHECK MH ***
      IF(MH.LE.0.) GO TO 1
      PH=PI-PPVF
C                CALC. RHO OF GAS
```

******** VENT ********

```
      CALL RHOLIQ(T,IG,RL)
      VU=V-ML/RL
      MHF = RHOG*VU
      DIF=ABS(MHF-MH)
      IF(DIF.GT.0.001) MH=MHF
    1 P=PI-DP
      MU=MPV+MH
      PB=0.5*(P+PI)
      MT=MU+ML
C
C     *** COMPUTE ENTHALPY OF VENT VAPOR ***
C
      PG=PPVF*PB/PI
      IF(MH.LE.0.) GO TO 3
      CALL FINTAB (NTBID(35))
      XTAB(1)=T
      XTAB(2)=PH
      HHI=MIPE(2,XTAB)
      PH=(PI-PPVF)*(PB/PI)
      XTAB(2)=PH
      HHV=MIPE(2,XTAB)
C                         CALC. RHO OF GAS
    3 CALL GSDNST (IG,T,PG,RG)
      CALL FINTAB (NTBID(36)+IG)
      XTAB(1)=RG
      XTAB(2)=PG
      HGI=MIPE(2,XTAB)
      HV=(MPV/MU)*HGI+(MH/MU)*HHV
C
C     *** COMPUTE NEW INTERNAL ENERGY FOR ASSUMED VENT MASS INCREMENT ***
C
      KTAB = 0
      IF (IG .EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
      CALL FINTAB (NTBID(37)+IG+KTAB)
      XTAB(1)=(ML+MPV)/V
      XTAB(2)=PPVF
      UI=MIPE(2,XTAB)*(ML+MPV)
C                         CALC. RHO OF GAS
      CALL GSZDNS (IG,T,PPVF,RHOG,ZPV)
      CALL ZFIND(T,PH,17,ZH)
      DEN=1.+(MH/MPV)*CMWR*(ZH/ZPV)
      PPVF=P/DEN
      MPVF = RHOG*VU
      DMV=(MH+MPV)*(1.-MPVF/MPV)
      DMVB=0.5*(DMVMAX+DMVMIN)
      IF(DMV.GE.DMVMAX) DMV=DMVB
      IF(DMV.LE.DMVMIN) DMV=DMVB
      UHI=0.
      IF(MH.LE.0.) GO TO 5
      PH=PI*(1.-1./DEN)
      UHI=MH*HHI-0.185*PH*VU
    5 UI=UI+UHI
   10 UFASS=UI+Q-DMV*HV
      LOOP=LOOP+1
C
C     *** COMPUTE PARTIAL PRESSURE OF PROPELLANT VAPOR ***
C
      MHF=MH*(1.-DMV/MU)
```

```
********   VENT   ********

              MPVF=MPV*(1.-DMV/MU)
      C
      C       *** COMPUTE INTERNAL ENERGY FOR THE NEW SATURATION CONDITIONS ***
      C
              KTAB = 0
              IF (IG .EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
              CALL FINTAB (NTBID(38)+IG+KTAB)
              XTAB(1)=(ML+MPVF)/V
              XTAB(2)=PPVF
              UF=MIPE(2,XTAB)*(ML+MPVF)
              TF=TSAT(PPVF,IG)
              IF(MHF.LE.0.) GO TO 15
              PHF=P-PPVF
              RH=MHF/VU
              CALL ZFIND(TF,PHF,17,ZH)
              PHF=FINDR(17)*RH*ZH*TF/144.
              CALL FINTAB (NTBID(39))
              XTAB(1)=TF
              XTAB(2)=PHF
              HHF=MIPE(2,XTAB)
              UHF=MHF*HHF-0.185*(P-PPVF)*VU
              IF(UHF.LT.0.) LOOP=20
      15      UF=UF+UHF
      C       ***  CHECKOUT WRITE STATEMENTS ARE PLACED HERE  ***
      C
      C       *** COMPARE UF AND UFASS ***
      C
              DIF=UFASS-UF
              DIFF=ABS(DIF)
              IF(DIFF.LE.1.) GO TO 60
      C
      C       *** REVISE VENT MASS INCREMENT ***
      C
              DMVC=DVMAX-DMV
              IF(DMVC.GT.DVTEST) GO TO 20
              DP=0.5*DP
              DP=0.5*DVMAX
              GO TO 70
      20      IF(DIF) 30,60,40
      30      DMVMAX=DMV
              IF(DMVMAX.GT.DVTEST) GO TO 50
              WRITE(6,1000) I,P,UFASS,UF,DMV
      1000    FORMAT(31HOS/R VENT ANOMALY AT ITERATION ,I2,13H DATA FOLLOW:/1X,
             14G13.8)
      40      DMVMIN=DMV
      50      DM=DMV-DIP*((DMV-DMVOLD)/(DIF-DIFOLD))
              DMVOLD=DMV
              DIFOLD=DIF
              DMV=DM
              IF(DMV.LE.DMVMAX.AND.DMV.GT.DMVMIN) GO TO 51
      53      DMV=0.5*(DMVMAX+DMVMIN)
      51      IF(LOOP.LT.20) GO TO 55
      C       ***  CHECKOUT WRITE STATEMENTS ARE PLACED HERE  ***
              GO TO 60
      55      GO TO 10
      C
      C       *** VENT MASS ITERATION HAS CONVERGED, RECOMPUTE MASSES AND
      C            CHECK FOR PRESSURE CONVERGENCE. ***
```

```
60    PI=P
      T=TF
      MH=MHF
      CALL RHOLIQ(T,IG,RL)
      CALL ZFIND(T,PPVF,IG,ZPV)
      RG=144.*PPVF/(ZPV*GC*T)
      RFP=(ML+MPVF)/V
      QUAL=(RL/RFP-1.)/(RL/RG-1.)
      IF(QUAL.GT.0.) GO TO 65
C     ***  CHECKOUT WRITE STATEMENTS ARE PLACED HERE  ***
      QUAL=0.01
      GO TO 66
65    IF(QUAL.LT.1.) GO TO 66
C     ***  CHECKOUT WRITE STATEMENTS ARE PLACED HERE  ***
      QUAL=1.
66     MTOT=ML+MPVF
      MPV=QUAL*MTOT
      ML=MTOT-MPV
C
      IF(DIAG(2,6HVENT-1)) WRITE (6,6010) MH,MPV,ML,T,UF,UFASS,PPVF,P
C
      DIF=P-PV
      ADIF=DIF-.001
      IF(ADIF.LE.0.) GO TO 80
      IF(DIF.GT.DP) GO TO 70
      DP=DIF
C
C     *****  DIAG SWITCH IS PLACED HERE  ******
C
70    CONTINUE
C
C     ***  CHECKOUT WRITE STATEMENTS ARE PLACED HERE  ***
C
   80 CONTINUE
C
C     *****  DIAG SWITCH IS PLACED HERE  ******
C
      JP = DIAG(1,6HVENT  )
C
      RETURN
C
 6000 FORMAT('+',14X,T23,'HEAT-IN',T37,'HE-ULL',T50,'PVAP-IN-ULL',T65,
     1 'LIQ-IN-TANK',T82,'TFLUID'/15X,5E15.8/T20,'VENT-PRES',T35,
     2 'INIT-PRES',T52,'TOT-VOL',T67,'IGAS',T80,'PRESSURE'/15X,3E15.8,
     3 6X,I3,6X,E15.8)
 6010 FORMAT('+',14X,T23,'-MH-',T37,'-MPV-',T52,'-ML-',T67,'-TFLUID-',
     1 /15X,4E15.8/T23,'-UF-',T37,'-UFASS-',T52,'-PPV-',T67,'--P--'/15X,
     2 4E15.8)
6020  FORMAT(39H0S/R VENT QUALITY ANOMALY AT ITERATION ,I2/
     1  T23,'QUAL',T37,'UF',T52,'ULF',T67,'UFG'/15X,4G13.8)
6030  FORMAT('0',14X,T23,'LOOP',T37,'UFASS',T52,'UF',T67,'MHF'/
     1  15X,I15,3G15.8/T21,'MPVF',T37,'TF',T52,'PPVF',T67,'PHF',
     2  T80,'DMV',/15X,5G15.8)
C
      END
```

******** SUBRØUTINE VGVS

```
C                 * * * * * * * * * * * * * ** * * * * * * * * * * *
C                 * ROUTINE NAME - MACH NUMBER COPUTATION           *
C                 * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                 * PROGRAMMER   - R. BOLLINGER 1963 102 26933 *
C                 * DATE CODED   - 3/17/70                          *
C                 * * * * * * * * * * * * * ** * * * * * * * * * * *
C
      SUBROUTINE VGVS(IDX,RHO,IGAS)
C
C         ***** EXPLANATION OF THE CALLING SEQUENCE
C               *
C               * IDX  - INDEX OF THE CONFIGURATION TABLE
C               *
C               * RHO  - DENSITY OF THE GAS
C               *
C         ***** IGAS - GAS NUMBER (SEE S.R. FINDR)
C
      INCLUDE CCNFIG
      INCLUDE CONST
C
      DATA IBLNK,IAST1,IAST6/'      ','*     ','******'/
C
C         ***** COMPUTE THE VELOCITY OF THE GAS
C
      MFLG(IDX) = IBLNK
      VG = 576. * WDOTN(IDX)/(PI * DIAM(IDX)**2 * RHO)
C
C         ***** COMPUTE THE VELOCITY OF SOUND IN GAS.
C
      CALL CSUBP(TEMP(IDX),PRES(IDX),IGAS,CPGAS)
      CVGAS = CSUBV(TEMP(IDX),PRES(IDX),IGAS)
C
      VS = SQRT(GRAVTY*CPGAS*FINDR(IGAS)*TEMP(IDX)/CVGAS)
C
C         ***** COMPUTE MACH NUMBER
C
      MACH(IDX) = VG/VS
C
C         ***** CHECH MACH NUMBER. FLAG MACH GREATER THAN .3 WITH ONE
C         ***** ASTRIK. FLAG MACH NUMBER GREATER THAN 1.0 WITH 6 ASTRIK
C
      IF(MACH(IDX) - 0.3) 40,40,10
   10 IF(MACH(IDX) - 1.0) 20,30,30
   20 MFLG(IDX) = IAST1
      GO TO 40
   30 MFLG(IDX) = IAST6
   40 RETURN
      END
```

******** FUNCTIØN VPN

```
      FUNCTION VPN(T)
      COMMON/CVPN/G(11) /CRPR/CR(3)
      COMMON /SCRH/ X(40)
C
C.... CALCULATE THE VAPOR PRESSURE
C
      TC=CR(3)
      A=G(11)
      T2=T*T
      T3=T*T2
      T4=T*T3
      T5=T*T4
      T6=T*T5
      X(1) = 1.0/T
      X(2) = 1.0
      X(3)=T
      X(4)=T2
      X(5)=T3
      X(6)=T4
      X(7)=T5
      X(8)=T6
      X(9) = ALOG(T)
      X(10)=(TC-T)**A
      P    = 0.0
      DO 1 I=1,10
    1 P=P+X(I)*G(I)
      P    = EXP(P)
      VPN=P
      RETURN
      END
```

********   FUNCTIØN VPNB

```
      FUNCTION VPNB(TB)
      T   = TB/1.8
      P=VPN(T)
      VPNB = P * 1.01325E+5/6.8947572E+3
      RETURN
      END
```

******** SUBRØUTINE VPRØP

```
      SUBROUTINE VPROP(T,P,D,K,H,S,U,Z)
      COMMON /RFPR/ RF(10)
C
C.... ROUTINE TO CALCULATE THE PROPERTIES OF THE VAPOR
C
C     K =1   INPUT IS T + D
C     K =2   INPUT IS T + P
C     K =3   INPUT IS T, P, + D
C
      IF(K.EQ.1)CALL PFND(T,D,P)
       IF(K.EQ.2)CALL DFND(T,P,D,ZI,0)
      HOTO=RF(1)
      SOTO=RF(2)
      RFST=RF(3)
      RFHT=RF(4)
      R   =RF(5)
      AK  =RF(6)
      FID=FING1(T,D)
      FIO  = FING1(T,0.0)
      F2D=FING2(T,D)
      F20  = FING2(T,0.0)
      SO=SOTO+CPSI(T)-CPSI(RFST)
      HO=HOTO+CPHI(T)-CPHI(RFHT)
      S    = SO - (R * ALOG(D*R*T)-FID + FIO) * AK
      H=HO+(T*(FID-FIO)+F2D-F20+P/D-R*T)*AK
      U=H-(P/D)*AK
      Z = P/(D*R*T)
      RETURN
      END
```

******** SUBRØUTINE VPRØPB

```
      SUBROUTINE VPROPB(TB,PB,DB,K,HB,SB,UB,ZB)
      COMMON /RFPR/RF(10)
      WT=RF(7)
      T   = TB/1.8
      P   = PB * 6.8947572E+3/1.01325E+5
      D   = DB * 453.59237E-3/(WT * 2.8316847E-2)
      CALL VPROP(T,P,D,K,H,S,U,Z)
      PB  = P * 1.01325E+5/6.8947572E+3
      DB  = D * WT * 2.8316847E-2/453.59237E-3
      HB  = H * 453.59237/(1.0543503E+3 * WT)
      UB  = U * 453.59237/(1.0543503E+3 * WT)
      SB  = S * 453.59237/(1.0543503E+3 * 1.8 * WT)
      ZB = Z
      RETURN
      END
```

******** SUBRØUTINE VSND

```
      SUBROUTINE VSND(T,P,D,K,W)
      COMMON /RFPR/ RF(10)
C
C.... ROUTINE TO CALCULATE THE SONIC VELOCITY FOR FOLLOWING INPUT OF K
C
C     K =1   INPUT IS T + P   RETURNS SONIC VELOCITY, W + D
C     K =2   INPUT IS T + D   RETURNS SONIC VELOCITY, W
C     K =3   INPUT IS T       RETURNS W, D, + P FOR SATURATED VAPOR
C     K =4   INPUT IS T       RETURNS W, D, + P FOR SATURATED LIQUID
C
      AK=RF(6)
      AM=RF(7)
      IF((K.GT.0).OR.(K.LT.5))GO TO 1
      WRITE(6,300)K
  300 FORMAT(' *** ERROR IN CALL VSND ***',/,
     .       '      K MUST EQUAL 1,2,3, OR 4',/,
     .       '      K = ',I10)
      RETURN
    1 IF(K.EQ.2)GO TO 3
      IF(K.GT.2)GO TO 2
       CALL DFND(T,P,D,Z1,0)
      GO TO 3
    2 P=VPN(T)
       IF(K.EQ.3)CALL DFND(T,P,D,Z1,2)
       IF(K.EQ.4)CALL DFND(T,P,D,Z1,1)
    3 CALL CPVTD(T,D,CP,CV)
      W = (CP/CV) * DPDD(T,D) * (AK * 1000.0/AM)
      IF(W.LE.0.0) GO TO 4
      W =  SQRT(W)
      RETURN
    4 CONTINUE
      W =  0.0
      RETURN
      END
```

********* SUBRØUTINE VSNDB

```
      SUBROUTINE VSNDB(TB,PB,DB,K,WB)
      COMMON /RFPR/ RF(10)
      WT=RF(7)
      T   = TB/1.8
      P   = PB * 6.8947572E+3/1.01325E+5
      D   = DB * 453.59237E-3/(WT * 2.8316847E-2)
      CALL VSND(T,P,D,K,W)
      WB  = W/3.048E-1
      RETURN
      END
```

******** SUBRØUTINE WFIND

```
      SUBROUTINE WFIND(T,W)
      DIMENSION A(20)
      DATA A/0.25084621E+3, 0.13509987E+3, 0.52785676E+2, 0.27676855E+2,
     1       0.39105321E+2, 0.65561323E+2, 0.80803587E+2, 0.70524212E+2,
     2       0.44784759E+2, 0.21252565E+2, 0.76797636E+1, 0.21368945E+1,
     3       0.45984335E+0, 0.76361463E-1, 0.96932862E-2, 0.92306915E-3,
     4       0.63811659E-4, 0.30229323E-5, 0.87755139E-7, 0.11770261E-8/
C
C.... ROUTINE TO CALCULATE W(IPTS-68(T68)) GIVEN TEMP
C
C.... CALCULATION DONE BY NEWTON'S METHOD
C
      ITERMX=100
      EPS = 1.0E-4
C
C.... CALC FIRST ESTIMATE
C
      W = 0.40713654E-2 * T - 0.11209346E+0
      DO 1 ITER=1,ITERMX
      DLW = ALOG(W)
C
C.... CALC T GIVEN EST OF W
C
      TT = 0.0
      DO 2 I=1,20
    2 TT=TT+A(I)*DLW**I
      TT = TT + 273.15
C
C.... CALC DT/DW
C
      DTDW = 0.0
      DO 3 I=1,20
      AI=I
    3 DTDW=DTDW+AI*A(I)*DLW**(I-1)/W
C
      DELTA=(T-TT)/DTDW
      W=W+DELTA
      IF(ABS(DELTA).LT.EPS) RETURN
    1 CONTINUE
      WRITE(6,100)DELTA
  100 FORMAT(' *** WFIND FAILED TO CONVERGE - DELTA =',G20.10)
      RETURN
      END
```

******** SUBRØUTINE WTACC

```
C                   * * * * * * * * * * * * * * * * * * * * * * *
C                   * ROUTINE NAME - ACCUMULATOR WEIGHTS        *
C                   *                DETERMINATION ROUTINE      *
C                   * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
C                   * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
C                   * DATE CODED   - 5/20/70                    *
C                   * REVISED      - JULY  1972                 *
C                   * PROGRAMMER   - J. MCKAY  D1943  201  45178 *
C                   * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE WTACC
C
      INCLUDE CACCUM
      INCLUDE CCNFIG
      INCLUDE CMATRL
C
      DIMENSION ADUMMY(2),ZERO(2)
C
      DATA ZERO / 0.,0./
C
C         ***** COMPUTE ACCUMULATOR WEIGHT AND SURFACE AREA.
C
      CALL TNKWTA (2,AVOL,ZERO,ADIAM,ATEMP,APRES,AMTYPE,ADUMMY,ACWT,AA,
     1 ACYHT)
      DO 10 II=1,2
      IAI = AITYPE(II)
C                   CALC. ACCUMULATOR INSULATION WEIGHT
      ACIWT(II) = NAOP(II)*AA(II)*RHOI(IAI)*AITHIK(II) / 12.
      ACWT(II) = NAOP(II)*ACWT(II)
      IDX = INDXAC(II)
      WEIGHT(IDX) = ACWT(II)
      WI(IDX) = ACIWT(II)
   10 CONTINUE
C                   PRINT ACCUMULATOR DATA
      CALL OTPACC
C
      RETURN
      END
```

```
      FUNCTION YLGINT(X,Y,N,XBAR,NPTS,$)                                YLGINT
C-----------------------------------------------------------------------YLGINT
C     LAGRANGE INTERPOLATION                                            YLGINT
C-----------------------------------------------------------------------YLGINT
C     ARGUMENT DEFINITION                                               YLGINT
C        X    -- ARRAY OF N INDEPENDENT VARIABLE VALUES(INPUT)          YLGINT
C        Y    -- ARRAY OF N DEPENDENT VARIABLE VALUES (INPUT)           YLGINT
C        N    -- NUMBER OF DATA POINTS (INPUT)                          YLGINT
C        XBAR -- INDEPENDENT VARIABLE VALUE FOR WHICH YLGINT IS THE     YLGINT
C                INTERPOLANT (INPUT)                                    YLGINT
C        NPTS -- NUMBER OF DATA POINTS TO BE USED IN INTERPOLATION(INPUTYLGINT
C        $    -- ERROR EXIT IF N IS LESS THAN 2, OR IF OVERFLOW IS      YLGINT
C                DETECTED.                                              YLGINT
C-----------------------------------------------------------------------YLGINT
C     REFERENCE                                                         YLGINT
C        K. S. KUNZ, NUMERICAL ANALYSIS, MCGRAW-HILL BOOK COMPANY, 1957.YLGINT
C-----------------------------------------------------------------------YLGINT
      DIMENSION X(N),Y(N)                                               YLGINT
C-----------------------------------------------------------------------YLGINT
C     CHECK ARGUMENTS N AND NPTS                                        YLGINT
C     IF NPTS IS OUT OF RANGE, USE ALL POINTS                           YLGINT
C-----------------------------------------------------------------------YLGINT
      IF(N.LT.2) RETURN 6                                               YLGINT
      IF(NPTS.GT.1.AND.NPTS.LT.N) GO TO 2                               YLGINT
      NPTS=N                                                            YLGINT
    1 JL=1                                                              YLGINT
      JH=NPTS                                                           YLGINT
      GO TO 9                                                           YLGINT
C-----------------------------------------------------------------------YLGINT
C     DETERMINE POSITION OF XBAR WITHIN X ARRAY                         YLGINT
C     UPON EXIT FROM LOOP 3, X(JH-1) LT XBAR LE X(JH)                   YLGINT
C-----------------------------------------------------------------------YLGINT
    2 DO 3 JH=1,N                                                       YLGINT
      IF(XBAR-X(JH)) 5,12,3                                             YLGINT
    3 CONTINUE                                                          YLGINT
    4 JH=N
      JL=N-NPTS+1
      GO TO 9                                                           YLGINT
C-----------------------------------------------------------------------YLGINT
C     DETERMINE WHICH POINTS USED IN INTERPOLATION                      YLGINT
C     THE POINTS CHOSEN ARE THE ONES WHOSE ABSCISSAS ARE CLOSEST TO XBARYLGINT
C     JL IS LOWEST SUBSCRIPT OF POINTS USED, JH IS HIGHEST SUBSCRIPT    YLGINT
C-----------------------------------------------------------------------YLGINT
    5 JL=JH                                                             YLGINT
      D2=X(JH)-XBAR                                                     YLGINT
    6 JL=JL-1                                                           YLGINT
      IF(JL.LE.1) GO TO 1                                               YLGINT
      IF(NPTS .EQ. 2) GO TO 99
      D1=XBAR-X(JL)                                                     YLGINT
    7 IF(JH-JL+1.EQ.NPTS) GO TO 8                                       YLGINT
      IF(XBAR-X(JL-1).LT.D2) GO TO 6                                    YLGINT
      JH=JH+1                                                           YLGINT
      IF(JH.GE.N) GO TO 4
      D2=X(JH)-XBAR                                                     YLGINT
      GO TO 7                                                           YLGINT
    8 INDCT=0                                                           YLGINT
      IF(JL.GT.1.AND.XBAR-X(JL-1).LT.D2) INDCT=-1
```

```
      IF(JH.LT.N.AND.X(JH+1)-XBAR.LT.D1) INDCT=1
      JL=JL+INDCT                                                       YLGINT
      JH=JH+INDCT                                                       YLGINT
C-----------------------------------------------------------------------YLGINT
C     DETERMINE INTERPOLATING POLYNOMIAL VALUE                          YLGINT
C     INNER LOOP COMPUTS PRODUCT                                        YLGINT
C     OUTER LOOP COMPUTES SUM                                           YLGINT
C     ERROR RETURN FOR DETECTED OVERFLOW                                YLGINT
C-----------------------------------------------------------------------YLGINT
    9 IF(NPTS .EQ. 2) GO TO 99
      YLGINT = 0.
      DO 11 K=JL,JH                                                     YLGINT
      PROD=1.                                                           YLGINT
      DO 10 J=JL,JH                                                     YLGINT
      IF(K.EQ.J) GO TO 10                                               YLGINT
      PROD=PROD*(XBAR-X(J))/(X(K)-X(J))                                 YLGINT
C                    TEST FOR OVERFLOW
      IF (PROD .GT. 1.E+20) RETURN 6
   10 CONTINUE                                                          YLGINT
      YLGINT=YLGINT+Y(K)*PROD                                           YLGINT
C                    TEST FOR OVERFLOW
      IF (YLGINT .GT. 1.E+20) RETURN 6
   11 CONTINUE                                                          YLGINT
      RETURN                                                            YLGINT
C-----------------------------------------------------------------------YLGINT
C     XBAR EQUAL TO AN ELEMENT OF THE X ARRAY                           YLGINT
C-----------------------------------------------------------------------YLGINT
   12 YLGINT=Y(JH)                                                      YLGINT
      RETURN                                                            YLGINT
C
C     FOR LINEAR INTERPOLATION USE THIS SIMPLIFIED FORM, MODIFIED BY
C     J.MCKAY  3-21-1972
C
   99 YLGINT = (Y(JH)-Y(JL))*(XBAR-X(JL))/(X(JH)-X(JL))+Y(JL)
      RETURN
      END                                                               YLGINT
```

******** FUNCTIØN YLNTRP

```
C
C                                   ROUTINE TO INTERPOLATE ON LOG-LOG
C                                   COORDINATES, FOR TABLES WHICH HAVE
C                                   BEEN INPUT LN(X),LNCY) FORM   (6-8-71)
C
      FUNCTION YLNTRP (NV,XVAL)
C                    NOT USED  --  DELETE
      YLNTRP = 1.0E+10
      RETURN
      END
```

******** SUBRØUTINE ZFIND

```
      SUBROUTINE ZFIND(T,P,N,V)
      DIMENSION G(3,17),S(17)
      DIMENSION A(17,6),TS(17)
      DATA (TS(K),K=1,16)
     1        /150., 30., 140., 30.,140.,190.,220.,190.,
     2         220.,300.,140.,370.,475.,480.,450.,450./
      DATA(G(1,I),I=1,17)/277.85,59.8,259.13,59.8,260.,343.2,387.,343.2
     1,387.,521.8,259.13,730.,776.4,1094.,749.5,1155.,9.37/
      DATA(G(2,I),I=1,17)/743.78,187.7,822.8,187.7,795.,673.1,719.,673.
     1 1,719.,581.,822.8,1652.,1470.,1696.,771.,1470.,33.82/
      DATA(G(3,I),I=1,17)/48.31,766.8,40.67,766.8,42.01,96.35,28.62,
     1 96.35,28.62,55.81,40.67,90.77,16.78,37.0,11.90,33.50,386.3/
      DATA S/5HLO2  ,5HLH2  ,5HLF2  ,5HLH2  ,5HFLOX ,5HCH4  ,5HOF2  ,
     1       5HCH4  ,5HOF2  ,5HB2H6 ,5HLF2  ,5HNH3  ,5HN2O4 ,5HA-50 ,
     2       5HCLF-5,5HMHF-5,5HHE   /
      DATA(A( 1,J),J=1,6)/.2142592E1,-.3228322E-1,.3563987E-3,
     1 -.1895669E-5,.4823166E-8,-.5002793E-11/
      DATA(A( 2,J),J=1,6)/-.4458459E1,.6350202,-.2863016E-1,
     1 .6260532E-3,-.6704223E-5,.2763409E-7/
      DATA(A( 3,J),J=1,6)/.19209203E1,-.4596045E-1,.7505833E-3,
     1 -.5522453E-5,.1900218E-7,-.2526448E-10/
      DATA(A( 4,J),J=1,6)/-.4458459E1,.6350202,-.2863016E-1,
     1 .6260532E-3,-.6704223E-5,.2763409E-7/
      DATA(A( 5,J),J=1,6)/.18476612E1,-.4146008E-1,.6702877E-3,
     1 -.4895455E-5,.1674002E-7,-.2217407E-10/
      DATA(A( 6,J),J=1,6)/.17005803E1,-.1669025E-1,.1536145E-3,
     1 -.6691577E-6,.13609972E-8,-.1135781E-11/
      DATA(A( 7,J),J=1,6)/.15219822E2,-.28012466,.22001021E-2,
     1 -.86119226E-5,.16835339E-7,-.1326415E-10    /
      DATA(A( 8,J),J=1,6)/.17005803E1,-.1669025E-1,.1536145E-3,
     1 -.6691577E-6,.13609972E-8,-.1135781E-11/
      DATA(A( 9,J),J=1,6)/.15219822E2,-.28012466,.22001021E-2,
     1 -.86119226E-5,.16835339E-7,-.1326415E-10    /
      DATA(A(10,J),J=1,6)/-.81449607E-1,.15604836E-1,-.91954274E-4,
     1 .27813987E-6,-.42739698E-9,.25170512E-12/
      DATA(A(11,J),J=1,6)/.19209203E1,-.4596045E-1,.7505833E-3,
     1 -.5522453E-5,.1900218E-7,-.2526448E-10/
      DATA(A(12,J),J=1,6)/.39233318E1,-.3456529E-1,.16306507E-3,
     1-.38452432E-6,.45572795E-9,-.21979859E-12/
      DATA(A(13,J),J=1,6)/.70122306E1,-.57413097E-1,.21967497E-3,
     1 -.42198721E-6,.40864988E-9,-.1606916E-12/
      DATA(A(14,J),J=1,6)/.86405843E1,-.72176161E-1,.2724231E-3,
     1 -.5137489E-6,.4844758E-9,-.18308062E-12/
      DATA(A(15,J),J=1,6)/.54858839E1,-.44806287E-1,.17789492E-3,
     1 -.35089558E-6,.34539726E-9,-.13832016E-12/
      DATA(A(16,J),J=1,6)/.49407545E1,-.39649956E-1,.15955648E-3,
     1 -.32115667E-6,.32356706E-9,-.13068156E-12/
C
      IF(P.GT.0.0.AND.T.GT.0.0) GO TO 666
      WRITE (6,777) P,T
  777 FORMAT(5X,'ENTERING ZFIND A PRESSURE OR A TEMPERATURE IS OUT OF RA
     1NGE'/T5,'PRESSURE =',E15.8,T30,'TEMPERATURE =',E15.8)
      CALL EXIT
  666 CONTINUE
C
      IF(N.EQ.17) GO TO 56
      IF(T.GT.650..OR.T.LT.25.) GO TO 35
```

```
*********    ZFIND    *********
      IF(T.GT.TS(N)+100..OR.T.LT.TS(N)) GO TO 56
C**** TEST TO SEE IF SAT.D COMP.Y Z APPLIES ******************************
      CALL PVAPOR(T,N,TRYP)
      TRY=TRYP-P
      IF(ABS(TRY).LT.5.) GO TO 55
   56 CONTINUE
C*****************************REDLICH-KWONG*****************AGO-GO***************0064
      VF=G(3,N)*T/(P*144.)
      IF(N.EQ.2.OR.N.EQ.4) GO TO 10
      AS=.4278*G(3,N)*G(3,N)/(G(2,N)*144.)*G(1,N)**2.5
      BS=.0867*G(3,N)*G(1,N)/(G(2,N)*144.)
      IN=0
      N2=0
      E=.00001
      V=VF
  199 Y=G(3,N)*T/(V-BS)-AS/(T**.5*V*(V+BS))-P*144.
      IF(ABS(Y).LT.E) GO TO 200
C*****************************FIRST DERIVITIVE OF REDLICH-KWONG RESPECT TO V*****0075
      YP=-G(3,N)
     1       *T/((V-BS)*(V-BS))+AS*(2.*V+BS)/(T**.5*V*V*(V+BS)*(V+BS))
      IN=IN+1
      N2=N2+1
      IF(N2.EQ.25) E=2.*E
      IF(N2.EQ.25) N2=0
     IF(IN.GT.1000)  GO TO 201
      V=V-Y/YP
      IF(V.GT.0.) GO TO 43
       OLDV=V+Y/YP
      DELTA=Y/YP
      FACTOR=.05
   44 V=OLDV-FACTOR*DELTA
      FACTOR=FACTOR*.9
      IF(V.LT.0.) GO TO 44
   43 CONTINUE
      GO TO 199
  200 CONTINUE
      GO TO 50
  201 CONTINUE
      WRITE(6,40) T,P,S(N),V,VF
   40 FORMAT(1X,34HREDLICH - KWONG      FLUNKED  T = ,F4.2,2X,4HP = ,
     1  F7.2,
     2  5H FOR ,A5,2X,4HV = ,E10.5,2X,8HRETURNED,E10.5)
      V=VF
      GO TO 50
   10 TT=T+0.5
      V=PTDENS(P     ,TT)
      V=1./V
   50 V=V/VF
      RETURN
   35 V=1.
      RETURN
   55 CONTINUE
      V=A(N,1)+A(N,2)*T+A(N,3)*T*T+A(N,4)*T*T*T+A(N,5)*T**4+A(N,6)*T**5
      RETURN
      END
```

********* FUNCTIØN ZGET

```
      FUNCTION ZGET(T,P,IGAS)
C
C     ROUTINE CALLS STEWARTS COMBINED O2-N2 PROPERTIES PROGRAM
C
      CALL ONPROP(T,P,DX,ENX,SX,UX,ZX,IGAS)
      ZGET = ZX
      RETURN
      END
```

Appendix B
THE CRYOGENIC INTEGRATED MATH MODEL (TCIMM)

PART II – PROGRAM FILE ELEMENT TABLE OF CONTENTS

The next several papes contain the program file table of contents, also know as the PRT,T index. The table is printed when called for by a ATPRT,T control card. The output contains the "element table," "procedure tables," and, if a ATPREP card has preceeded the PRT card, the "entry point table."

The column headings given at the beginning of the element table have the following meanings:

D-FLAG	– an asterisk means that the entry is deleted from the file.
NAME	– name of symbolic/relocatable/absolute element.
VERSION	– version of element.
TYPE	– if the element is symbolic, the processor which created it is indicated.
DATE, TIME	– time that element was added to the file.
SEQUENCE NO.	– position of the element in the file. This is sequentially issued as elements are added to the file.
PRE-SIZE	– for relocatable elements, the preamble length is given in sectors (28 words per sector).
TEXT-SIZE	– this is the text size in sectors.
CYCLE WORD	– the cycle word is broken up into three separate parameters; starting from left to right, they are: (1) the number of cycles the system will maintain (2) the number of the most current cycles (absolute scale) (3) the number of cycles currently being maintained.
LOCATION	– refers to the sector position relating to the start of the file (1792 is the base).

It should be noted that the entry point table is fugitive in the sense that it must be recreated each time a change is made in the program, and is subject to the following constraints:

(1) Destroyed when an update is made to any element in a program file.

(2) Destroyed when program file is put on magnetic tape.

(3) Is not re-established when file is copied from tape to drum.

(4) Contains externalized labels.

(5) Is created by the ATPREP statement which will prepare or re-establish an entry point table for a specified program file.

TABLE ØF CØNTENTS - FILE TCIMM.

D	NAME	VERSION	TYPE	DATE	TIME	SEQ #	SIZE-PRE	TEXT	(CYCLE	WORD)	PSRMODE	LOCATION
	CACCUM		FOR PROC	13 MAR 73	01:51:49	1		14	1	0		1792
	CAPU		FOR PROC	13 MAR 73	01:51:50	2		50	1	0		1806
	CCNFIG		FOR PROC	13 MAR 73	01:51:51	3		43	1	0		1856
	CCNTRL		FOR PROC	13 MAR 73	01:51:52	4		10	1	0		1899
	CDCYCL		FOR PROC	13 MAR 73	01:51:52	5		10	1	0		1909
	CENG		FOR PROC	13 MAR 73	01:51:54	6		10	1	0		1919
	CFLRAT		FOR PROC	13 MAR 73	01:51:54	7		3	1	0		1929
	CFLUID		FOR PROC	13 MAR 73	01:51:54	8		1	1	0		1932
	CHEX		FOR PROC	13 MAR 73	01:51:55	9		22	1	0		1933
	CHTX		FOR PROC	13 MAR 73	01:51:56	10		4	1	0		1955
	CHSORC		FOR PROC	13 MAR 73	01:51:57	11		13	1	0		1959
	CIOUNT		FOR PROC	13 MAR 73	01:51:57	12		5	1	0		1972
	CKEYS		FOR PROC	13 MAR 73	01:51:58	13		1	1	0		1977
	CMATRL		FOR PROC	13 MAR 73	01:51:58	14		17	1	0		1978
	CMOTOR		FOR PROC	13 MAR 73	01:51:59	15		2	1	0		1995
	CONST		FOR PROC	13 MAR 73	01:52:00	16		2	1	0		1997
	CPAGE		FOR PROC	13 MAR 73	01:52:00	17		8	1	0		1999
	CPUMP		FOR PROC	13 MAR 73	01:52:01	18		20	1	0		2007
	CSYSWT		FOR PROC	13 MAR 73	01:52:01	19		6	1	0		2027
	CTAB		FOR PROC	13 MAR 73	01:52:02	20		10	1	0		2033
	CTABA		FOR PROC	13 MAR 73	01:52:02	21		2	1	0		2043
	CTANK		FOR PROC	13 MAR 73	01:52:03	22		19	1	0		2045
	CTURBN		FOR PROC	13 MAR 73	01:52:03	23		13	1	0		2064
	DUMMY		FOR PROC	13 MAR 73	01:52:04	24		3	1	0		2077
	SPUMP		FOR PROC	13 MAR 73	01:52:04	25		2	1	0		2080
	TABLOK		FOR PROC	13 MAR 73	01:52:06	26		1	1	0		2082
	TANKWT		FOR PROC	13 MAR 73	01:52:06	27		3	1	0		2083
	ACCRES		RELOCATABLE	13 MAR 73	01:52:09	28	1	3				2086
	ACCRES		FOR SYMB	13 MAR 73	01:52:09	29		7	5	0		2090
	ACQWT		RELOCATABLE	13 MAR 73	01:52:13	30	2	3				2097
	ACQWT		FOR SYMB	13 MAR 73	01:52:13	31		10	5	0		2102
	ALPHAB		RELOCATABLE	13 MAR 73	01:52:15	32	1	1				2112
	ALPHAB		FOR SYMB	13 MAR 73	01:52:15	33		3	5	0		2114
	APUFLO		RELOCATABLE	13 MAR 73	01:52:20	34	3	15				2117
	APUFLO		FOR SYMB	13 MAR 73	01:52:20	35		31	5	0		2135
	APUSUP		RELOCATABLE	13 MAR 73	01:52:49	36	5	100				2166

APUSUP	FOR SYMB	13 MAR 73	01:52:49	37		156	5	0	1	2271
BETAB	RELOCATABLE	13 MAR 73	01:52:52	38	1	3				2427
BETAB	FOR SYMB	13 MAR 73	01:52:52	39		5	5	0	1	2431
CFTW	RELOCATABLE	13 MAR 73	01:52:55	40	1	9				2436
CFTW	FOR SYMB	13 MAR 73	01:52:55	41		7	5	0	1	2446
COMFLO	RELOCATABLE	13 MAR 73	01:53:18	42	2	15				2453
COMFLO	FOR SYMB	13 MAR 73	01:53:18	43		25	5	0	1	2470
CONSUM	RELOCATABLE	13 MAR 73	01:54:05	44	2	2				2495
CONSUM	FOR SYMB	13 MAR 73	01:54:06	45		3	5	0	1	2499
CPHI	RELOCATABLE	13 MAR 73	01:54:10	46	2	6				2502
CPHI	FOR SYMB	13 MAR 73	01:54:11	47		5	5	0	1	2510
CPIG	RELOCATABLE	13 MAR 73	01:54:13	48	2	5				2515
CPIG	FOR SYMB	13 MAR 73	01:54:13	49		6	5	0	1	2522
CPSI	RELOCATABLE	13 MAR 73	01:54:16	50	2	6				2528
CPSI	FOR SYMB	13 MAR 73	01:54:16	51		5	5	0	1	2536
CPVTDB	RELOCATABLE	13 MAR 73	01:54:18	52	1	3				2541
CPVTDB	FOR SYMB	13 MAR 73	01:54:18	53		3	5	0	1	2545
CPVTD	RELOCATABLE	13 MAR 73	01:54:21	54	2	8				2548
CPVTD	FOR SYMB	13 MAR 73	01:54:21	55		6	5	0	1	2558
CRYCON	RELOCATABLE	13 MAR 73	01:54:25	56	3	9				2564
CRYCON	FOR SYMB	13 MAR 73	01:54:25	57		15	5	0	1	2576
CSPF21	RELOCATABLE	13 MAR 73	01:54:27	58	1	5				2591
CSPF21	FOR SYMB	13 MAR 73	01:54:27	59		4	5	0	1	2597
CSUBP	RELOCATABLE	13 MAR 73	01:54:30	60	2	8				2601
CSUBP	FOR SYMB	13 MAR 73	01:54:30	61		11	5	0	1	2611
CSUBPV	RELOCATABLE	13 MAR 73	01:54:32	62	2	4				2622
CSUBPV	FOR SYMB	13 MAR 73	01:54:32	63		5	5	0	1	2628
CSUBPI	RELOCATABLE	13 MAR 73	01:54:35	64	1	11				2633
CSUBPI	FOR SYMB	13 MAR 73	01:54:36	65		17	5	0	1	2645
CSUBV	RELOCATABLE	13 MAR 73	01:54:38	66	2	6				2662
CSUBV	FOR SYMB	13 MAR 73	01:54:38	67		9	5	0	1	2670
DATAN2	RELOCATABLE	13 MAR 73	01:54:43	68	3	16				2679
DATAN2	FOR SYMB	13 MAR 73	01:54:43	69		36	5	0	1	2698
DATA02	RELOCATABLE	13 MAR 73	01:54:48	70	3	16				2734
DATA02	FOR SYMB	13 MAR 73	01:54:48	71		35	5	0	1	2753
DCALC	RELOCATABLE	13 MAR 73	01:55:01	72	1	15				2788
DCALC	FOR SYMB	13 MAR 73	01:55:02	73		16	5	0	1	2804
DENSON	RELOCATABLE	13 MAR 73	01:55:08	74	2	3				2820
DENSON	FOR SYMB	13 MAR 73	01:55:08	75		5	5	0	1	2825
DFNDB	RELOCATABLE	13 MAR 73	01:55:10	76	1	3				2830
DFNDB	FOR SYMB	13 MAR 73	01:55:10	77		3	5	0	1	2834
DFND	RELOCATABLE	13 MAR 73	01:55:12	78	2	11				2837
DFND	FOR SYMB	13 MAR 73	01:55:13	79		9	5	0	1	2850
DIAG	RELOCATABLE	13 MAR 73	01:55:15	80	1	5				2859
DIAG	FOR SYMB	13 MAR 73	01:55:15	81		11	5	0	1	2865
DPDDB	RELOCATABLE	13 MAR 73	01:55:16	82	1	3				2876
DPDDB	FOR SYMB	13 MAR 73	01:55:16	83		3	5	0	1	2880
DPDD	RELOCATABLE	13 MAR 73	01:55:21	84	2	15				2883
DPDD	FOR SYMB	13 MAR 73	01:55:21	85		12	5	0	1	2900
DPDTB	RELOCATABLE	13 MAR 73	01:55:22	86	1	2				2912
DPDTB	FOR SYMB	13 MAR 73	01:55:22	87		2	5	0	1	2915
DPDT	RELOCATABLE	13 MAR 73	01:55:26	88	2	12				2917
DPDT	FOR SYMB	13 MAR 73	01:55:27	89		11	5	0	1	2931
DPDTVP	RELOCATABLE	13 MAR 73	01:55:31	90	2	6				2942
DPDTVP	FOR SYMB	13 MAR 73	01:55:31	91		5	5	0	1	2950
DSATL	RELOCATABLE	13 MAR 73	01:55:39	92	2	4				2955
DSATL	FOR SYMB	13 MAR 73	01:55:39	93		5	5	0	1	2961
DSATV	RELOCATABLE	13 MAR 73	01:55:40	94	2	4				2966

DSATV	FOR SYMB	13 MAR 73	01:55:41	95		5	5	0	I	2972
ENGINE	RELOCATABLE	13 MAR 73	01:56:11	96	3	17				2977
ENGINE	FOR SYMB	13 MAR 73	01:56:11	97		22	5	0	I	2997
FINDR	RELOCATABLE	13 MAR 73	01:56:13	98	1	3				3019
FINDR	FOR SYMB	13 MAR 73	01:56:13	99		2	5	0	I	3023
FING1	RELOCATABLE	13 MAR 73	01:56:16	100	2	14				3025
FING1	FOR SYMB	13 MAR 73	01:56:16	101		12	5	0	I	3041
FING2	RELOCATABLE	13 MAR 73	01:56:20	102	2	14				3053
FING2	FOR SYMB	13 MAR 73	01:56:20	103		11	5	0	I	3069
FING3	RELOCATABLE	13 MAR 73	01:56:24	104	2	14				3080
FING3	FOR SYMB	13 MAR 73	01:56:24	105		12	5	0	I	3096
FINTAB	RELOCATABLE	13 MAR 73	01:56:27	106	2	9				3108
FINTAB	FOR SYMB	13 MAR 73	01:56:27	107		12	5	0	I	3119
FLORAT	RELOCATABLE	13 MAR 73	01:56:32	108	3	22				3131
FLORAT	FOR SYMB	13 MAR 73	01:56:33	109		28	5	0	I	3156
FLODEQ	RELOCATABLE	13 MAR 73	01:56:34	110	1	3				3184
FLODEQ	FOR SYMB	13 MAR 73	01:56:34	111		8	5	0	I	3188
GASGEN	RELOCATABLE	13 MAR 73	01:57:10	112	2	10				3196
GASGEN	FOR SYMB	13 MAR 73	01:57:10	113		15	5	0	I	3208
GETCON	RELOCATABLE	13 MAR 73	01:57:14	114	2	4				3223
GETCON	FOR SYMB	13 MAR 73	01:57:14	115		8	5	0	I	3229
GOMTRY	RELOCATABLE	13 MAR 73	01:57:18	116	3	28				3237
GOMTRY	FOR SYMB	13 MAR 73	01:57:18	117		14	5	0	I	3268
HEATEX	RELOCATABLE	13 MAR 73	01:57:30	118	3	67				3282
HEATEX	FOR SYMB	13 MAR 73	01:57:31	119		89	5	0	I	3352
HEXELC	RELOCATABLE	13 MAR 73	01:57:36	120	2	14				3441
HEXELC	FOR SYMB	13 MAR 73	01:57:36	121		24	5	0	I	3457
HTLEAK	RELOCATABLE	13 MAR 73	01:57:42	122	1	1				3481
HTLEAK	FOR SYMB	13 MAR 73	01:57:42	123		2	5	0	I	3483
HPTCP	RELOCATABLE	13 MAR 73	01:57:44	124	1	2				3485
HPTCP	FOR SYMB	13 MAR 73	01:57:45	125		3	5	0	I	3488
HPTCV	RELOCATABLE	13 MAR 73	01:57:46	126	1	2				3491
HPTCV	FOR SYMB	13 MAR 73	01:57:46	127		3	5	0	I	3494
HPTGAM	RELOCATABLE	13 MAR 73	01:57:48	128	1	2				3497
HPTGAM	FOR SYMB	13 MAR 73	01:57:48	129		3	5	0	I	3500
HPW	RELOCATABLE	13 MAR 73	01:57:50	130	1	1				3503
HPW	FOR SYMB	13 MAR 73	01:57:50	131		4	5	0	I	3505
HVAP	RELOCATABLE	13 MAR 73	01:57:55	132	1	8				3509
HVAP	FOR SYMB	13 MAR 73	01:57:55	133		5	5	0	I	3518
HYENTH	RELOCATABLE	13 MAR 73	01:58:10	134	1	89				3523
HYENTH	FOR SYMB	13 MAR 73	01:58:11	135		137	5	0	I	3613
INTAB	RELOCATABLE	13 MAR 73	01:59:15	136	2	57				3750
INTAB	FOR SYMB	13 MAR 73	01:59:16	137		66	5	0	I	3809
LIQRES	RELOCATABLE	13 MAR 73	01:59:19	138	1	4				3875
LIQRES	FOR SYMB	13 MAR 73	01:59:19	139		11	5	0	I	3880
LOCATE	RELOCATABLE	13 MAR 73	01:59:23	140	2	9				3891
LOCATE	FOR SYMB	13 MAR 73	01:59:23	141		13	5	0	I	3902
LPROPB	RELOCATABLE	13 MAR 73	01:59:26	142	1	6				3915
LPROPB	FOR SYMB	13 MAR 73	01:59:26	143		4	5	0	I	3922
LPROP	RELOCATABLE	13 MAR 73	01:59:29	144	2	13				3926
LPROP	FOR SYMB	13 MAR 73	01:59:30	145		8	5	0	I	3941
LWEGHT	RELOCATABLE	13 MAR 73	02:00:22	146	2	16				3949
LWEGHT	FOR SYMB	13 MAR 73	02:00:23	147		24	5	0	I	3967
MATHAX	RELOCATABLE	13 MAR 73	02:00:41	148	4	29				3991
MATHAX	FOR SYMB	13 MAR 73	02:00:42	149		30	5	0	I	4024
MIPE	RELOCATABLE	13 MAR 73	02:00:47	150	2	21				4054
MIPE	FOR SYMB	13 MAR 73	02:00:47	151		28	5	0	I	4077
NIENTH	RELOCATABLE	13 MAR 73	02:00:49	152	1	2				4105

NIENTH	FOR SYMB	13 MAR 73	02:00:49	153		2	5	0	1	4108
ONPROP	RELOCATABLE	13 MAR 73	02:00:52	154	2	7				4110
ONPROP	FOR SYMB	13 MAR 73	02:00:52	155		8	5	0	1	4119
OUTPUT	RELOCATABLE	13 MAR 73	02:01:29	156	2	21				4127
OUTPUT	FOR SYMB	13 MAR 73	02:01:29	157		13	5	0	1	4150
OXENTH	RELOCATABLE	13 MAR 73	02:01:34	158	1	2				4163
OXENTH	FOR SYMB	13 MAR 73	02:01:34	159		2	5	0	1	4166
PAGE	RELOCATABLE	13 MAR 73	02:01:43	160	2	7				4168
PAGE	FOR SYMB	13 MAR 73	02:01:43	161		31	5	0	1	4177
PARPMP	RELOCATABLE	13 MAR 73	02:01:52	162	3	27				4208
PARPMP	FOR SYMB	13 MAR 73	02:01:52	163		43	5	0	1	4238
PFNDB	RELOCATABLE	13 MAR 73	02:01:57	164	1	3				4281
PFNDB	FOR SYMB	13 MAR 73	02:01:57	165		2	5	0	1	4285
PFND	RELOCATABLE	13 MAR 73	02:02:01	166	2	12				4287
PFND	FOR SYMB	13 MAR 73	02:02:01	167		11	5	0	1	4301
PHIB	RELOCATABLE	13 MAR 73	02:02:04	168	1	2				4312
PHIB	FOR SYMB	13 MAR 73	02:02:04	169		3	5	0	1	4315
PROPB	RELOCATABLE	13 MAR 73	02:02:10	170	1	6				4318
PROPB	FOR SYMB	13 MAR 73	02:02:11	171		4	5	0	1	4325
PROP	RELOCATABLE	13 MAR 73	02:02:15	172	2	20				4329
PROP	FOR SYMB	13 MAR 73	02:02:15	173		21	5	0	1	4351
PSATH	RELOCATABLE	13 MAR 73	02:02:18	174	1	10				4372
PSATH	FOR SYMB	13 MAR 73	02:02:18	175		18	5	0	1	4383
PTDENS	RELOCATABLE	13 MAR 73	02:02:36	176	1	72				4401
PTDENS	FOR SYMB	13 MAR 73	02:02:37	177		70	5	0	1	4474
PTHEAT	RELOCATABLE	13 MAR 73	02:02:52	178	1	34				4544
PTHEAT	FOR SYMB	13 MAR 73	02:02:52	179		68	5	0	1	4579
PHTHON	RELOCATABLE	13 MAR 73	02:03:16	180	2	4				4647
PHTHON	FOR SYMB	13 MAR 73	02:03:16	181		6	5	0	1	4653
PVAPOR	RELOCATABLE	13 MAR 73	02:03:34	182	1	16				4659
PVAPOR	FOR SYMB	13 MAR 73	02:03:34	183		7	5	0	1	4676
RHOLIQ	RELOCATABLE	13 MAR 73	02:03:37	184	1	11				4683
RHOLIQ	FOR SYMB	13 MAR 73	02:03:37	185		6	5	0	1	4695
SPHSEG	RELOCATABLE	13 MAR 73	02:03:41	186	2	19				4701
SPHSEG	FOR SYMB	13 MAR 73	02:03:42	187		13	5	0	1	4722
SPHTDA	RELOCATABLE	13 MAR 73	02:03:53	188	1	75				4735
SPHTDA	FOR SYMB	13 MAR 73	02:03:54	189		83	5	0	1	4811
STOCON	RELOCATABLE	13 MAR 73	02:03:57	190	1	3				4894
STOCON	FOR SYMB	13 MAR 73	02:03:57	191		7	5	0	1	4898
TBOIL	RELOCATABLE	13 MAR 73	02:06:55	192	1	2				4905
TBOIL	FOR SYMB	13 MAR 73	02:06:55	193		2	5	0	1	4908
TCOND	RELOCATABLE	13 MAR 73	02:07:07	194	1	17				4910
TCOND	FOR SYMB	13 MAR 73	02:07:07	195		22	5	0	1	4928
TEL	RELOCATABLE	13 MAR 73	02:07:35	196	2	9				4950
TEL	FOR SYMB	13 MAR 73	02:07:36	197		15	5	0	1	4961
TEMP	RELOCATABLE	13 MAR 73	02:07:45	198	1	18				4976
TEMP	FOR SYMB	13 MAR 73	02:07:45	199		15	5	0	1	4995
THETAB	RELOCATABLE	13 MAR 73	02:07:48	200	1	2				5010
THETAB	FOR SYMB	13 MAR 73	02:07:48	201		3	5	0	1	5013
THKWTG	RELOCATABLE	13 MAR 73	02:08:09	202	2	16				5016
THKWTG	FOR SYMB	13 MAR 73	02:08:10	203		24	5	0	1	5034
TKGEOM	RELOCATABLE	13 MAR 73	02:08:28	204	3	20				5058
TKGEOM	FOR SYMB	13 MAR 73	02:08:28	205		20	5	0	1	5081
TMELTB	RELOCATABLE	13 MAR 73	02:08:53	206	1	2				5101
TMELTB	FOR SYMB	13 MAR 73	02:08:53	207		2	5	0	1	5104
TMELT	RELOCATABLE	13 MAR 73	02:08:56	208	1	5				5106
TMELT	FOR SYMB	13 MAR 73	02:08:57	209		4	5	0	1	5112
TNKWTA	RELOCATABLE	13 MAR 73	02:09:08	210	2	34				[illegible]

TNKWTA	FOR SYMB	13 MAR 73	02:09:08	211		46	5	0	1	5152
TRAC	RELOCATABLE	13 MAR 73	02:09:15	212	1	2				5198
TRAC	FOR SYMB	13 MAR 73	02:09:15	213		2	5	0	1	5201
TSAT	RELOCATABLE	13 MAR 73	02:09:22	214	1	11				5203
TSAT	FOR SYMB	13 MAR 73	02:09:22	215		7	5	0	1	5215
TSATH	RELOCATABLE	13 MAR 73	02:09:32	216	1	10				5222
TSATH	FOR SYMB	13 MAR 73	02:09:33	217		18	5	0	1	5233
TSIZEI	RELOCATABLE	13 MAR 73	02:09:39	218	3	17				5251
TSIZEI	FOR SYMB	13 MAR 73	02:09:39	219		27	5	0	1	5271
TSTART	RELOCATABLE	13 MAR 73	02:09:41	220	1	2				5298
TSTART	FOR SYMB	13 MAR 73	02:09:41	221		3	5	0	1	5301
TURBN	RELOCATABLE	13 MAR 73	02:09:44	222	2	6				5304
TURBN	FOR SYMB	13 MAR 73	02:09:44	223		9	5	0	1	5312
TVPB	RELOCATABLE	13 MAR 73	02:09:46	224	1	2				5321
TVPB	FOR SYMB	13 MAR 73	02:09:46	225		2	5	0	1	5324
TVP	RELOCATABLE	13 MAR 73	02:09:48	226	2	8				5326
TVP	FOR SYMB	13 MAR 73	02:09:49	227		9	5	0	1	5336
VARNAM	RELOCATABLE	13 MAR 73	02:09:54	228	6	1				5345
VARNAM	FOR SYMB	13 MAR 73	02:09:54	229		6	5	0	1	5352
VENT	RELOCATABLE	13 MAR 73	02:10:01	230	2	44				5358
VENT	FOR SYMB	13 MAR 73	02:10:01	231		47	5	0	1	5404
VGVS	RELOCATABLE	13 MAR 73	02:10:03	232	2	6				5451
VGVS	FOR SYMB	13 MAR 73	02:10:04	233		12	5	0	1	5459
VPNB	RELOCATABLE	13 MAR 73	02:10:06	234	1	2				5471
VPNB	FOR SYMB	13 MAR 73	02:10:06	235		2	5	0	1	5474
VPN	RELOCATABLE	13 MAR 73	02:10:07	236	2	6				5476
VPN	FOR SYMB	13 MAR 73	02:10:07	237		5	5	0	1	5484
VPROPB	RELOCATABLE	13 MAR 73	02:10:10	238	1	6				5489
VPROPB	FOR SYMB	13 MAR 73	02:10:10	239		4	5	0	1	5496
VPROP	RELOCATABLE	13 MAR 73	02:10:12	240	2	9				5500
VPROP	FOR SYMB	13 MAR 73	02:10:12	241		6	5	0	1	5511
VSNDB	RELOCATABLE	13 MAR 73	02:10:14	242	1	3				5517
VSNDB	FOR SYMB	13 MAR 73	02:10:14	243		3	5	0	1	5521
VSND	RELOCATABLE	13 MAR 73	02:10:16	244	2	9				5524
VSND	FOR SYMB	13 MAR 73	02:10:16	245		8	5	0	1	5535
WFIND	RELOCATABLE	13 MAR 73	02:10:19	246	1	8				5543
WFIND	FOR SYMB	13 MAR 73	02:10:19	247		9	5	0	1	5552
WTACC	RELOCATABLE	13 MAR 73	02:10:22	248	2	4				5561
WTACC	FOR SYMB	13 MAR 73	02:10:22	249		10	5	0	1	5567
YLGINT	RELOCATABLE	13 MAR 73	02:10:25	250	1	16				5577
YLGINT	FOR SYMB	13 MAR 73	02:10:25	251		42	5	0	1	5594
YLNTRP	RELOCATABLE	13 MAR 73	02:10:27	252	1	1				5636
YLNTRP	FOR SYMB	13 MAR 73	02:10:27	253		3	5	0	1	5638
ZFIND	RELOCATABLE	13 MAR 73	02:10:31	254	2	35				5641
ZFIND	FOR SYMB	13 MAR 73	02:10:32	255		33	5	0	1	5678
ZGET	RELOCATABLE	13 MAR 73	02:10:40	256	1	2				5711
ZGET	FOR SYMB	13 MAR 73	02:10:40	257		2	5	0	1	5714
APUSUB	RELOCATABLE	14 MAR 73	22:09:49	258	4	28				5716
APUSUB	FOR SYMB	14 MAR 73	22:09:49	259		44	5	0	1	5748
CONTRL	RELOCATABLE	14 MAR 73	22:12:32	260	3	10				5792
CONTRL	FOR SYMB	14 MAR 73	22:12:32	261		13	5	0	1	5805
CNAMES	FOR PROC	28 MAR 73	00:05:08	262		3	1	0	1	5818
TANK	RELOCATABLE	28 MAR 73	00:14:24	263	4	161				5821
TANK	FOR SYMB	28 MAR 73	00:14:25	264		157	5	0	1	5986
CECLSS	FOR PROC	28 MAR 73	10:31:37	265		19	1	0	1	6143
CFUEL	FOR PROC	28 MAR 73	10:31:39	266		13	1	0	1	6162
CMPCAL	RELOCATABLE	28 MAR 73	10:32:04	267	6	90				6175
CMPCAL	FOR SYMB	28 MAR 73	10:32:04	268		126	5	0	1	6271

COMPIL	RELOCATABLE	28 MAR 73	10:32:52	269	6	168				6397
COMPIL	FOR SYMB	28 MAR 73	10:32:52	270		137	5	0	1	6571
STODTA	RELOCATABLE	28 MAR 73	10:35:43	271	4	75				6708
STODTA	FOR SYMB	28 MAR 73	10:35:44	272		103	5	0	1	6787
LSSCMP	RELOCATABLE	29 MAR 73	04:24:13	273	4	52				6890
LSSCMP	FOR SYMB	29 MAR 73	04:24:14	274		71	5	0	1	6946
OTRTNS	RELOCATABLE	29 MAR 73	04:26:02	275	7	141				7017
OTRTNS	FOR SYMB	29 MAR 73	04:26:03	276		127	5	0	1	7165
ECLSS	RELOCATABLE	06 APR 73	13:32:02	277	6	190				7292
ECLSS	FOR SYMB	06 APR 73	13:32:04	278		218	5	0	1	7488
FUELCL	RELOCATABLE	06 APR 73	13:51:52	279	5	217				7706
FUELCL	FOR SYMB	06 APR 73	13:51:56	280		260	5	0	1	7928
HEXF21	RELOCATABLE	06 APR 73	13:52:00	281	1	31				8188
HEXF21	FOR SYMB	06 APR 73	13:52:00	282		15	5	0	1	8220
NEXT AVAILABLE LOCATION-										8235

ASSEMBLER PROCEDURE TABLE EMPTY

COBOL PROCEDURE TABLE EMPTY

FORTRAN PROCEDURE TABLE

D NAME	LOCATION	LINK	D NAME	LOCATION	LINK	D NAME	LOCATION	LINK
CACCUM	50178	1	CAPU	50570	2	CCNFIG	51970	3
CCNTRL	53174	4	CDCYCL	53454	5	CECLSS	172006	265
CENG	53734	6	CFLRAT	54014	7	CFLUID	54098	8
CFUEL	172538	266	CHEX	54126	9	CHSORC	54854	11
CHTX	54742	10	CIOUNT	55218	12	CKEYS	55358	13
CMATRL	55386	14	CMOTOR	55862	15	CNAMES	162906	262
CONST	55918	16	CPAGE	55974	17	CPUMP	56198	18
CSYSWT	56758	19	CTAB	56926	20	CTABA	57206	21
CTANK	57262	22	CTURBN	57794	23	DUMMY	58158	24
SPUMP	58242	25	TABLOK	58298	26	TANKWT	58326	27

ENTRY POINT TABLE EMPTY

Appendix B
THE CRYOGENIC INTEGRATED MATH MODEL (TCIMM)

PART III – CROSS REFERENCE OF PROGRAM FILE

It is often of interest to know which subprograms call a specific routine in a given program sequence. This kind of information for program file TCIMM is presented in the following pages. An explanation of the XREF processes which generates the cross reference listing is given below.

XREF (Cross Reference Listing of Relocatable Elements): The XREF processor generates a cross reference listing of all entry points and undefined symbols in the specified program file that has been PACKed and PREP'D. The names of the relocatable elements are listed alphabetically. Beside each element name, the names of the element entry points are listed. Beside each entry point name, the names of all relocatable elements in the program file which reference this entry point are listed. An element entry point is the result of an assembly or compilation and specifies the location at which execution of the program element commences. A compiled FORTRAN V subroutine or function has one entry point corresponding to the name of the subroutine or function. An undefined symbol (or external reference) is the result of a subroutine call or reference to an array not contained within the element. Any external FORTRAN reference creates an undefined symbol.

```
@XREF*XREF.XREF

 ACCRES         01 (000037) (ACCRES) ,CRYCON
 ACQWT          01 (000034) (ACQWT) ,CRYCON
 AFUNC          01 (000446) (TKGEOM) ,TNKWTA
 ALPHAB         01 (000007) (ALPHAB)
 APUFLO         01 (000337) (APUFLO) ,CONSUM
 APUSUB         01 (000707) (APUSUB) ,CRYCON
 APUSUP         01 (003175) (APUSUP) ,CRYCON
 ARACYL         01 (000617) (GOMTRY) ,TKGEOM,TNKWTA
 AREAFR         01 (000660) (GOMTRY) ,TKGEOM
 ARSPHR         01 (000727) (GOMTRY) ,TKGEOM,TNKWTA
 BETAB          01 (000056) (BETAB) ,ECLSS,FUELCL
 CFTW           01 (000131) (CFTW) ,CMPCAL,LSSCMP
 CMPCAL         01 (002673) (CMPCAL) ,CRYCON
 COMFLO         01 (000263) (COMFLO) ,CMPCAL,LSSCMP
 COMPIL         01 (005142) (COMPIL) ,CONTRL
 CONE           01 (000245) (GOMTRY)
 CONSUM         01 (000027) (CONSUM) ,CRYCON
 CPHI           01 (000103) (CPHI) ,VPROP
 CPIG           01 (000103) (CPIG) ,CPVTD
 CPSI           01 (000113) (CPSI) ,VPROP
 CPVTD          01 (000157) (CPVTD) ,VSND,CPVTDB
 CPVTDB         01 (000040) (CPVTDB) ,PHTHON,CSUBPV
 CRYCON         01 (000202) (CRYCON) ,CONTRL
 CSPF21         01 (000063) (CSPF21) ,FUELCL
 CSUBP          01 (000151) (CSUBP) ,HEXELC,VGVS,APUSUB,COMFLO,APUSUP,ECLSS,FUELCL
 CSUBPV         01 (000056) (CSUBPV) ,CSUBV,CSUBP
 CSUBPI         01 (000115) (CSUBPI) ,GASGEN,FLORAT,APUSUB,CMPCAL,APUSUP,APUFLO
 CSUBV          01 (000113) (CSUBV) ,VGVS,TANK,COMFLO,ECLSS,APUSUP,FUELCL
 CYLHED         01 (000364) (SPHSEG) ,TKGEOM,TNKWTA
 CYLNDR         01 (000306) (GOMTRY) ,TKGEOM
 CYLSPH         01 (000347) (GOMTRY) ,TKGEOM
 CYNSPH         01 (000451) (SPHSEG) ,TKGEOM
 DATAN2         01 (000257) (DATAN2) ,PHTHON,ONPROP,DENSON,CSUBPV,BETAB
 DATAO2         01 (000265) (DATAO2) ,PHTHON,ONPROP,DENSON,CSUBPV,BETAB
 DCALC          01 (000274) (DCALC) ,DFND
 DENSON         01 (000052) (DENSON) ,LSSCMP,ECLSS,APUSUP,FUELCL
 DFND           01 (000211) (DFND) ,PROP,LPROP,VPROP,VSND,DFNDB
 DFNDB          01 (000034) (DFNDB) ,DENSON,CSUBPV
 DIAG           01 (000062) (DIAG) ,MIPE,LWEGHT,LOCATE,TSIZEI,VENT,GETCON,FLORAT,FINTAB,ENGINE,TANK,CRYCON,COMFLO

 DPDD           01 (000327) (DPDD) ,VSND,DPDDB,CPVTD
 DPDDB          01 (000034) (DPDDB) ,PHTHON,BETAB
 DPDT           01 (000271) (DPDT) ,DPDTB,CPVTD
 DPDTB          01 (000027) (DPDTB) ,PHTHON,BETAB
 DPDTVP         01 (000101) (DPDTVP) ,LPROP,TVP
 DSATL          01 (000060) (DSATL) ,DFND
 DSATV          01 (000064) (DSATV) ,DFND
 ECLSS          01 (004406) (ECLSS) ,CRYCON
 ELIPSG         01 (000324) (SPHSEG) ,TKGEOM
 ENGINE         01 (000310) (ENGINE) ,CONSUM
 ENTHOH         01 (000560) (MATHAX) ,HEATEX,FLORAT
 FDNSTY         01 (000576) (MATHAX) ,THKWTG,TSIZEI
 FINDR          01 (000014) (FINDR) ,MATHAX,VENT,VGVS,TANK,COMFLO,APUSUP
 FING1          01 (000322) (FING1) ,LPROP,VPROP
 FING2          01 (000314) (FING2) ,LPROP,VPROP
 FING3          01 (000321) (FING3) ,CPVTD
```

```
FINTAB    01 (000163) (FINTAB) ,PARPMP,MATHAX,LWEGHT,THKWTG,HEXELC,VENT,HEATEX,ENGINE,TANK,CMPCAL,ECLSS,APUSUP
                      ,FUELCL,APUFLO
FLODEQ    01 (000041) (FLODEQ) ,COMFLO
FLORAT    01 (000531) (FLORAT) ,CONSUM
FRCONE    01 (000422) (GOMTRY) ,TKGEOM
FRHEAD    01 (000422) (SPHSEG) ,TKGEOM
FUELCL    01 (005355) (FUELCL) ,CRYCON
GASGEN    01 (000211) (GASGEN) ,CMPCAL,APUSUP
GETCON    01 (000053) (GETCON) ,CMPCAL,LSSCMP,OTRTNS
GSDNST    01 (000623) (MATHAX) ,VENT,APUSUB,TANK,CMPCAL,FUELCL,ACCRES
GSZDNS    01 (000650) (MATHAX) ,VENT,TANK
HEATEX    01 (001742) (HEATEX) ,TANK,CMPCAL,APUSUP
HEXELC    01 (000263) (HEXELC) ,ECLSS
HEXF21    01 (000751) (HEXF21) ,FUELCL
HFUNC     01 (000504) (TKGEOM) ,TNKWTA
HPTCP     01 (000014) (HPTCP) ,CSUBP
HPTCV     01 (000014) (HPTCV) ,CSUBV
HPTGAM    01 (000014) (HPTGAM) ,COMFLO
HPW       01 (000006) (HPW)
HSPHER    01 (000471) (GOMTRY) ,TKGEOM
HTLEAK    01 (000006) (HTLEAK)
HVAP      01 (000110) (HVAP)
HYENTH    01 (000621) (HYENTH) ,MATHAX,APUSUB,APUSUP,FUELCL
INTAB     01 (001510) (INTAB) ,CONTRL
LIQRES    01 (000063) (LIQRES) ,CRYCON
LOCAT     01 (000167) (LOCATE) ,MIPE
LPROP     01 (000254) (LPROP) ,PROP,LPROPB,CPVTD
LPROPB    01 (000073) (LPROPB)
LSSCMP    01 (001452) (LSSCMP) ,ECLSS
LWEGHT    01 (000353) (LWEGHT) ,CMPCAL,LSSCMP
MIPE      01 (000506) (MIPE) ,PARPMP,THKWTG,MATHAX,LWEGHT,HEXELC,VENT,HEATEX,ENGINE,TANK,CMPCAL,ECLSS,FUELCL
                      ,APUSUP,APUFLO
NIENTH    01 (000022) (NIENTH) ,ECLSS
OAPUSB    01 (003773) (OTRTNS) ,APUSUB
OAPUSP    01 (003766) (OTRTNS) ,APUSUP
ONPROP    01 (000146) (ONPROP) ,OXENTH,NIENTH,ZGET
OPAPUF    01 (003761) (OTRTNS) ,APUFLO
OPTPOW    01 (003722) (OTRTNS) ,ECLSS
OTPACC    01 (003742) (OTRTNS) ,WTACC
OTPACQ    01 (003745) (OTRTNS) ,ACQWT
OTPFLT    01 (003753) (OTRTNS) ,FLORAT
OTPFLX    01 (003756) (OTRTNS) ,FLORAT
OTPHEX    01 (003711) (OTRTNS) ,CMPCAL,APUSUP
OTPHXE    01 (003717) (OTRTNS) ,ECLSS
OTPHXF    01 (003714) (OTRTNS) ,FUELCL
OTPPMP    01 (003725) (OTRTNS) ,CMPCAL
OTPTRB    01 (003730) (OTRTNS) ,CMPCAL
OTPTSZ    01 (003733) (OTRTNS) ,TSIZEI
OTPWSM    01 (003750) (OTRTNS) ,CRYCON
OTUNIT    01 (000512) (OUTPUT) ,CONTRL
OUTPA     01 (000425) (OUTPUT) ,OTRTNS
OUTPF     01 (000344) (OUTPUT) ,OTRTNS
OUTPFI    01 (000371) (OUTPUT) ,OTRTNS
OUTPI     01 (000402) (OUTPUT) ,OTRTNS
OUTPW     01 (000450) (OUTPUT) ,OTRTNS
OXENTH    01 (000022) (OXENTH) ,MATHAX,APUSUB,ECLSS,APUSUP,FUELCL
PAGE      01 (000121) (PAGE) ,INTAB,ENGINE,CONTRL,DIAG,TANK,CMPCAL,COMPIL,LSSCMP,OTRTNS,ECLSS,FUELCL
PARPMP    01 (000600) (PARPMP) ,CMPCAL
```

```
PFND        01 (000251) (PFND) ,PFNDB,LPROP,VPROP.DCALC,CPVTD
PFNDB       01 (000030) (PFNDB)
PHIB        01 (000012) (PHIB) ,PHTHON
PHTHON      01 (000100) (PHTHON) ,ECLSS,APUSUP,FUELCL
PROP        01 (000374) (PROP) ,PROPB
PROPB       01 (000073) (PROPB) ,ONPROP
PSATH       01 (000122) (PSATH) ,APUSUB
PTDENS      01 (000551) (PTDENS) ,ZFIND
PTHEAT      01 (000671) (PTHEAT) ,HPTCP,HPTCV,HPTGAM
PUMPEF      01 (000522) (MATHAX) ,PARPMP
PVAPOR      01 (000316) (PVAPOR) ,ZFIND
RHOLIQ      01 (000203) (RHOLIQ) ,MATHAX,VENT,FLORAT,APUSUB,TANK,CMPCAL
RVMIPE      01 (000521) (MIPE)
SPACE       01 (000507) (OUTPUT) ,OTRTNS
SPHERE      01 (000544) (GOMTRY) ,TNKWTA
SPHSEG      01 (000266) (SPHSEG) ,TNKWTA
STOCON      01 (000040) (STOCON) ,COMPIL
STODTA      01 (000006) (STODTA) ,CONTRL
TANK        01 (003716) (TANK) ,CRYCON
TBOIL       01 (000014) (TBOIL) ,HVAP
TCOND       01 (000371) (TCOND) ,APUSUB,APUSUP,ECLSS,FUELCL
TCRCAL      01 (000527) (MATHAX) ,HEATEX
TCRCLC      01 (000700) (MATHAX) ,HEATEX
TCRLOW      01 (000735) (MATHAX) ,HEATEX
TCRRAZ      01 (000746) (MATHAX) ,HEATEX
TEL         01 (000161) (TEL) ,MIPE
TEMP        01 (000321) (TEMP) ,TMELT
THETAB      01 (000012) (THETAB) ,PHTHON
THKWTG      01 (000323) (THKWTG) ,TNKWTA
TMELT       01 (000077) (TMELT) ,TMELTB
TMELTB      01 (000021) (TMELTB)
TNKWTA      01 (001057) (TNKWTA) ,TSIZEI,WTACC
TRAC        01 (000013) (TRAC)
TSAT        01 (000234) (TSAT) ,HEATEX,VENT,TANK
TSATH       01 (000122) (TSATH)
TSIZEI      01 (000416) (TSIZEI) ,CRYCON
TSTART      01 (000014) (TSTART) ,HVAP
TURBN       01 (000105) (TURBN) ,CMPCAL
TVP         01 (000136) (TVP) ,TVPB
TVPB        01 (000020) (TVPB)
VARNAM      01 (000004) (VARNAM)
VENT        01 (001105) (VENT) ,TANK
VFUNC       01 (000410) (TKGEOM) ,TNKWTA
VGVS        01 (000106) (VGVS) ,CMPCAL,LSSCMP
VPN         01 (000104) (VPN) ,PROP,LPROP,TVP,VPNB,VSND,DFND
VPNB        01 (000020) (VPNB)
VPROP       01 (000157) (VPROP) ,PROP,LPROP,VPROPB
VPROPB      01 (000073) (VPROPB)
VSND        01 (000145) (VSND) ,VSNDB
VSNDB       01 (000036) (VSNDB)
WFIND       01 (000124) (WFIND) ,TEMP
WOUACL      01 (000726) (MATHAX) ,HEATEX
WTACC       01 (000062) (WTACC) ,CRYCON
YLGINT      01 (000405) (YLGINT) ,TEL
YLNTRP      01 (000007) (YLNTRP)
ZFIND       01 (000436) (ZFIND) ,MATHAX,VENT,APUSUB,TANK,APUSUP,FUELCL
ZGET        01 (000022) (ZGET) ,FUELCL,APUSUP,ECLSS,APUSUB
**DONE**
```